

Основы построения защищенных баз данных

Ваша команда по спасению компьютерной безопасности

30 мая 2020 г.

Самый надежный в мире алгоритм
Всегда делать исключительно то, что горит
В самый прекрасный последний момент
Ведь для самых важных дел в принципе лучше времени нет

Anacondaz - Факап

Содержание

| | | |
|----------|--|-----------|
| 1 | Концепция безопасности БД | 2 |
| 1.0.1 | Понятие безопасности БД | 2 |
| 1.0.2 | Угрозы безопасности БД: общие и специфичные | 4 |
| 1.0.3 | Требования безопасности БД | 8 |
| 1.0.4 | Защита от несанкционированного доступа | 11 |
| 1.0.5 | Защита от вывода | 12 |
| 1.0.6 | Целостность БД | 12 |
| 1.0.7 | Аудит | 13 |
| 1.0.8 | Многоуровневая защита | 14 |
| 1.0.9 | Типы контроля безопасности: потоковый, контроль вывода, контроль доступа | 15 |
| 2 | Механизмы, поддерживающие высокую готовность | 17 |
| 2.1 | Средства, поддерживающие высокую готовность | 17 |
| 2.2 | Оперативное администрирование | 21 |
| 2.3 | Функциональная насыщенность СУБД | 22 |
| 2.4 | Системы, обладающие свойством высокой готовности | 24 |
| 3 | Защита данных в распределенных системах | 25 |
| 3.1 | Распределенные вычислительные среды | 25 |
| 3.2 | Угрозы безопасности распределенных СУБД | 26 |
| 3.3 | Распределенная обработка данных | 34 |
| 3.4 | Протоколы фиксации | 36 |
| 3.5 | Тиражирование данных | 39 |
| 3.6 | Интеграция БД и Internet | 41 |

| | | |
|----------|---|-----------|
| 4 | Безопасность в статистических БД | 41 |
| 4.1 | Общие сведения | 41 |
| 4.2 | Классификация | 42 |
| 4.3 | Угрозы статистических БД | 43 |
| 4.4 | Защита в статистических БД | 43 |

1 Концепция безопасности БД

1.0.1 Понятие безопасности БД

Для того чтобы иметь общую точку старта нам придется дать пару определений, я постараюсь быстро разобраться с обязательной копипастой и перейти к делу. Итак:

База данных¹ – это организованная коллекция данных, обычно хранящихся и доступных в электронном виде из компьютерной системы. Там, где базы данных более сложны, они часто разрабатываются с использованием формальных методов проектирования и моделирования.

Система управления базами данных (СУБД)¹ – это программное обеспечение, которое взаимодействует с конечными пользователями, приложениями и самой базой данных для сбора и анализа данных. Программное обеспечение СУБД дополнительно включает в себя основные средства, предоставляемые для администрирования базы данных. Общая сумма базы данных, СУБД и связанных приложений может называться «системой базы данных». Часто термин «база данных» также используется для обозначения любой СУБД, системы баз данных или приложения, связанного с базой данных.

Но если вас вдруг спросят, то смело отвечайте:

Согласно **гражданскому кодексу** Российской Федерации (часть четвертая) от 18.12.2006 N 230-ФЗ (ред. от 18.07.2019), базой данных является представленная в объективной форме совокупность самостоятельных материалов (статей, расчетов, нормативных актов, судебных решений и иных подобных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ).

И если с просто базами данных все более-менее понятно, то в тот момент, когда нам приходится говорить о вопросах безопасности, все превращается в классическую задачу о двух стульях: как надо и как в законе написано. (Ну или пафосно перефразируя) Вопросы информационной безопасности баз данных целесообразно рассматривать с двух взаимодополняющих позиций (Лихоносов А. Г. 2011)²:

- оценочные стандарты, направленные на классификацию информационных систем и средств их защиты по требованиям безопасности

¹Нагло скопипизженно с **вечно загнивающей**

²Мне стыдно указывать ресурс, где я взял этот кусок, так что пусть будет **pornhub**

- технические спецификации, регламентирующие различные аспекты реализации средств защиты

Попытка защищать все и сразу обречена на провал, так что довольно логичным кажется сосредоточиться на обеспечении безопасности четырех уровней информационной системы (Лихоносов А. Г. 2011) :

1. уровня прикладного программного обеспечения, отвечающего за взаимодействие с пользователем
2. уровня системы управления базами данных, обеспечивающего хранение и обработку данных информационной системы
3. уровня операционной системы, отвечающего за функционирование СУБД и иного прикладного программного обеспечения
4. уровня среды доставки, отвечающего за взаимодействие информационных серверов и потребителей информации

Теперь уже можно ввести недостающие определения. Я вижу как вы соскучились по определениям³.

Если БД рассматривать только как совокупность данных, то можно использовать следующее определение:

- **Безопасность информации [данных]:** Состояние защищенности информации [данных], при котором обеспечены ее [их] конфиденциальность, доступность и целостность.
- **Информационная система (ИС)** – система, предназначенная для хранения, поиска и обработки информации, и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию.

Если БД рассматривать как информационную систему, то можно использовать следующие определения:

- **Безопасность ИС (БД)** можно определить как состояние защищенности ИС от угроз ее нормальному функционированию. Под защищенностью понимается наличие средств ИС и методов их применения, обеспечивающих снижение или ликвидацию негативных последствий, связанных с реализацией угроз. Изложенный подход к определению понятия безопасности ИС предполагает, что перечень и содержание угроз достаточно хорошо определены и достаточно стабильны во времени.
- **Безопасность ИС (БД)** можно определить как свойство системы адаптироваться к агрессивным проявлениям среды, в которой функционирует система, обеспечивающее поддержку на экономически оправданном уровне характеристики качества системы. В сформулированном определении основной

³А эти определения я взял с прошлого года

акцент делается не на перечне и содержании угроз, нейтрализация которых обеспечивается, а на особую характеристику качества системы. При этом основной критерий качества ИС является экономическим, т.е. оценка средств и методов обеспечения безопасности осуществляется на основе затрат на реализацию механизмов безопасности и потенциальных выгод от недопущения ущерба, связанного с целенаправленным или случайным агрессивным проявлением среды.

Ну и не одно введение не может обойтись без заклинания

Проблема обеспечения безопасности автоматизированных информационных систем может быть определена как решение трех взаимосвязанных задач по реализации требуемого уровня:

- **конфиденциальности** – обеспечения пользователям доступа только к данным, для которых пользователь имеет явное или неявное разрешение на доступ
- **целостности** – обеспечения защиты от преднамеренного или непреднамеренного изменения информации или процессов ее обработки
- **доступности** – обеспечения возможности авторизованным в системе пользователям доступа к информации в соответствии с принятой технологией

1.0.2 Угрозы безопасности БД: общие и специфичные

Интуитивное понимание угрозы безопасности можно сформулировать как нарушение велико-
лепной тройки: Трус, Балбес и Бывалый: конфиденциальность, целостность, доступность. Для формального диалога можно использовать что-то около⁴:

Угрозой информационной безопасности автоматизированной информационной системе (АИС) назовем возможность воздействия на информацию, обрабатываемую в системе, приводящего к искажению, уничтожению, копированию, блокированию доступа к информации, а также возможность воздействия на компоненты информационной системы, приводящего к утрате, уничтожению или сбою функционирования носителя информации или средства управления программно-аппаратным комплексом системы.

Для того, чтобы разобраться во всем зоопарке перечисленных воздействий на систему нам придется все это дело классифицировать. Для удобства будем сразу смотреть на это со стороны обобщающего, то есть по источнику воздействия. В этом контексте они довольно естественно разбиваются на внутренние и внешние. Для описания внешних угроз необходимо учитывать объекты воздействия. (Под объектами воздействия понимаются объекты, которые могут подвергнуться атакам или могут стать причиной их возникновения.) (Утебов Данияр Рашидович 2008)

⁴Нагло взято [отсюда](#)

Внешними дестабилизирующими факторами, создающими угрозы безопасности функционированию систем баз данных и СУБД, являются:

- умышленные, деструктивные действия лиц с целью искажения, уничтожения или хищения программ, данных и документов системы, причиной которых являются нарушения информационной безопасности защищаемого объекта
- искажения в каналах передачи информации, поступающей от внешних источников, циркулирующих в системе и передаваемой потребителям, а также недопустимые значения и изменения характеристик потоков информации из внешней среды и внутри системы
- сбои и отказы в аппаратуре вычислительных средств
- вирусы и иные деструктивные программные элементы, распространяемые с использованием систем телекоммуникаций, обеспечивающих связь с внешней средой или внутренние коммуникации распределенной системы баз данных
- изменения состава и конфигурации комплекса взаимодействующей аппаратуры системы за пределы, проверенные при тестировании или сертификации системы

Внутренними источниками угроз безопасности баз данных и СУБД являются:

- системные ошибки при постановке целей и задач проектирования автоматизированных информационных систем и их компонент, допущенные при формулировке требований к функциям и характеристикам средств обеспечения безопасности системы
- ошибки при определении условий и параметров функционирования внешней среды, в которой предстоит использовать информационную систему и, в частности, программно-аппаратные средства защиты данных
- ошибки и несанкционированные действия пользователей, административного и обслуживающего персонала в процессе эксплуатации системы
- недостаточная эффективность используемых методов и средств обеспечения информационной безопасности в штатных или особых условиях эксплуатации системы

Если у вас разбегаются глаза, это нормально – здесь перечислены атаки на всех уровнях . Что за уровни? Напоминаю:

- На уровне сети
 - Activex-объект
 - Интерфейсы: OLE DB, ADO, ODBC, JDBC
 - Протоколы: TCP/IP, IPX/SPX, Named Pipes, Multiprotocol
 - Рабочие станции
 - Серверы
 - Маршрут
 - URL

- На уровне ОС
 - Аппаратное обеспечение
 - Программное обеспечение
 - Файлы базы данных
 - Файлы журнала транзакций
 - Файлы резервного копирования
 - Transact-SQL, PLSQL
 - Службы: MSSQLServer, SQLServerAgent, TNSListener и т. д.
- На уровне БД
 - Пользователи
 - Роли
 - Роли приложения
 - Диаграммы
 - Представления
 - Таблицы
 - Хранимые процедуры
 - Определения по умолчанию
 - Правила
 - Функции
 - Тип данных

Дальше, также легко и непринужденно можно разбить все атаки на СУБД на:

1. атаки на уровне ОС
2. атаки на уровне сети
3. атаки на уровне БД

Атаки на ОС, в которых функционирует СУБД, возникают гораздо чаще, так как защитить ОС гораздо сложнее, чем СУБД. Это обусловлено тем, что число различных типов защищаемых объектов в современных ОС может достигать нескольких десятков, а число различных типов защищаемых информационных потоков – нескольких сотен. Возможность практической реализации той или иной атаки на ОС в значительной мере определяется архитектурой и конфигурацией ОС. Тем не менее существуют атаки, которые могут быть направлены практически на любые ОС:

1. Кража ключевой информации (паролей)
2. Подбор пароля
3. Сканирование жестких дисков компьютера

4. Превышение полномочий
5. Атаки класса «Отказ в обслуживании»

Наиболее опасные атаки на СУБД исходят из сетей. На уровне сетевого программного обеспечения возможны следующие атаки на СУБД:

1. Прослушивание канала
2. Перехват пакетов на маршрутизаторе
3. Создание ложного маршрутизатора
4. Навязывание пакетов
5. Атаки класса «Отказ в обслуживании»

Теперь спускаемся на уровень самой БД. Для простоты восприятия разобьем все кучкам угроз конфиденциальности, целостности и доступности (Опять же, согласно (Утебов Данияр Рашидович 2008)):

- К угрозам конфиденциальности информации можно отнести следующие :
 1. Инъекция SQL. Во многих приложениях используется динамический SQL – формирование SQL-предложений кодом программы путем конкатенации строк и значений параметров. Зная структуру базы данных, злоумышленник может либо выполнить хранимую программу в запросе, либо закомментировать «легальные» фрагменты SQL-кода, внедрив, например, конструкцию UNION, запрос которой возвращает конфиденциальные данные. В последнее время злоумышленник может использовать специальные программы, автоматизирующие процесс реализации подобных угроз.
 2. Логический вывод на основе функциональных зависимостей. Пусть дана схема отношения: $R(A_1, \dots, A_n)$. Пусть $U = \{A_1, \dots, A_n\}$, X, Y – подмножества из U . X функционально определяет Y , если в любом отношении r со схемой $R(A_1, \dots, A_n)$ не могут содержаться два кортежа с одинаковыми значениями атрибутов из X и с различными из Y . В этом случае имеет место функциональная зависимость, обозначаемая $X \Rightarrow Y$. В реальных БД при наличии сведений о функциональных зависимостях злоумышленник может вывести конфиденциальную информацию при наличии доступа только к части отношений, составляющих декомпозированное отношение.
 3. Логический вывод на основе ограничений целостности. Для кортежей отношений в реляционной модели данных можно задать ограничения целостности – логические условия, которым должны удовлетворять атрибуты кортежей. При этом ограничение целостности может быть задано в виде предиката на всем множестве атрибутов кортежа. В случае попытки изменить данные в таблице, СУБД автоматически вычисляет значение этого предиката, и в зависимости от его истинности операция разрешается или отвергается. Многократно изменяя данные и анализируя реакцию системы, злоумышленник может получить те сведения, к которым у него нет непосредственного доступа. К этому виду угроз можно отнести также анализ значений первичных/вторичных ключей.

4. Использование оператора UPDATE для получения конфиденциальной информации. В некоторых стандартах SQL пользователь, не обладая привилегией на выполнение оператора SELECT, мог выполнить оператор UPDATE со сложным логическим условием. Так как после выполнения оператора UPDATE сообщается, сколько строк он обработал, фактически пользователь мог узнать, существуют ли данные, удовлетворяющие этому условию.
- К угрозам целостности информации, специфические для СУБД можно отнести следующие :
 1. С помощью SQL-операторов UPDATE, INSERT и DELETE можно изменить данные в СУБД. Опасность заключается в том, что пользователь, обладающий соответствующими привилегиями, может модифицировать все записи в таблице.
 - К угрозам доступности для СУБД можно отнести следующие:
 1. Использование свойств первичных и внешних ключей. В первую очередь отнесем сюда свойство уникальности первичных ключей и наличие ссылочной целостности. В том случае, если используются натуральные, а не генерируемые системой значения первичных ключей, может создаться такая ситуация, когда в таблицу невозможно будет вставить новые записи, так как там уже будут записи с такими же значениями первичных ключей. Если в БД поддерживается ссылочная целостность, можно организовать невозможность удаления родительских записей, умышленно создав подчиненные записи.
 2. Блокировка записей при изменении. Заблокировав записи или всю таблицу, злоумышленник может на значительное время сделать ее недоступной для обновления.
 3. Загрузка системы бессмысленной работой. Злоумышленник может выполнить запрос, содержащий декартово произведение двух больших отношений. Мощность декартового произведения двух отношений мощности N_1 и N_2 равна $N_1 \cdot N_2$. Это означает, что при выдаче злоумышленником запроса вида `SELECT * FROM Tab1, Tab1 ORDER BY 1`, где мощность отношения (количество строк в таблице Tab1) $N_1 = 10000$, мощность результирующего отношения будет $N = N_1^2 = 10000^2$. Вычисление соединения и сортировка результирующего отношения потребуют значительных ресурсов системы и отрицательно скажутся на производительности операций других пользователей.
 4. Использование разрушающих программных средств. Например, атака типа «троянский конь» – запуск пользователями программ, содержащих код, выполняющий определенные действия, внедренный туда злоумышленником.

1.0.3 Требования безопасности БД

Если сильно постараться то, на основании угроз можно выделить следующий список требований к безопасности БД⁵:

- Функционирование в доверенной среде. Под доверенной средой следует понимать инфраструктуру предприятия и ее защитные механизмы, обусловленные политиками безопасности. Таким образом, речь идет о функционировании СУБД в соответствии с правилами безопасности, применяемыми и ко всем прочим системам предприятия

⁵За оригиналом [сюда](#)

- Организация физической безопасности файлов данных. Требования к физической безопасности файлов данных СУБД в целом не отличаются от требований, применяемых к любым другим файлам пользователей и приложений
- Организация безопасной и актуальной настройки СУБД. Данное требование включает в себя общие задачи обеспечения безопасности, такие как своевременная установка обновлений, отключение неиспользуемых функций или применение эффективной политики паролей
- Безопасность пользовательского ПО. Сюда можно отнести задачи построения безопасных интерфейсов и механизмов доступа к данным
- Безопасная организация и работа с данными. Вопрос организации данных и управления ими является ключевым в системах хранения информации. В эту область входят задачи организации данных с контролем целостности и другие, специфичные для СУБД проблемы безопасности. Фактически эта задача включает в себя основной объем зависящих от данных уязвимостей и защиты от них

Но настало время отставить логику в сторону и поговорить на эльфийском о юридических требованиях. В целом, в России пытаются регламентировать эту отрасль, но в основном это басни про «кругом враги», «безопасная безопасность» и «импортозамещать до потери пульса». Даже «вертикаль власти» не забыли [(Мысев Алексей Эдуардович 2019)].⁶

Для беглого ознакомления достаточно заглянуть в [ФЗ «Об информации, информационных технологиях и о защите информации»](#) и в [Указ президента «О мерах по обеспечению информационной безопасности Российской Федерации при использовании информационно-телекоммуникационных сетей международного информационного обмена»](#). Хотя что-то более-менее содержательное начинается с [требований ФСТЭК](#).

Итак, согласно ФСТЭК у автоматизированная система управления, имеет многоуровневую структуру :

1. уровень операторского (диспетчерского) управления (верхний уровень)
2. уровень автоматического управления (средний уровень)
3. уровень ввода (вывода) данных исполнительных устройств (нижний (полевой) уровень)

В автоматизированной системе управления объектами защиты являются:

1. информация (данные) о параметрах (состоянии) управляемого (контролируемого) объекта или процесса
2. программно-технический комплекс, включающий технические средства, программное обеспечение, а также средства защиты информации

Принимаемые организационные и технические меры защиты информации:

- должны обеспечивать доступность обрабатываемой в автоматизированной системе управления информации (исключение неправомерного блокирования информации), ее целостность (исключение неправомерного уничтоже-

⁶Если очень хочется, то можно преисполниться вот [этой монографией](#)

ния, модифицирования информации), а также, при необходимости, конфиденциальность (исключение неправомерного доступа, копирования, предоставления или распространения информации)

- должны соотноситься с мерами по промышленной, физической, пожарной, экологической, радиационной безопасности, иными мерами по обеспечению безопасности автоматизированной системы управления и управляемого (контролируемого) объекта и (или) процесса
- не должны оказывать отрицательного влияния на штатный режим функционирования автоматизированной системы управления.

Организационные и технические меры защиты информации, реализуемые в автоматизированной системе управления в рамках ее системы защиты, в зависимости от класса защищенности, угроз безопасности информации, используемых технологий и структурно-функциональных характеристик автоматизированной системы управления и особенностей ее функционирования должны обеспечивать:

- идентификацию и аутентификацию (ИАФ)
- управление доступом (УПД)
- ограничение программной среды (ОПС)
- защиту машинных носителей информации (ЗНИ)
- аудит безопасности (АУД)
- антивирусную защиту (АВЗ)
- предотвращение вторжений (компьютерных атак) (СОВ)
- обеспечение целостности (ОЦЛ)
- обеспечение доступности (ОДТ)
- защиту технических средств и систем (ЗТС)
- защиту информационной (автоматизированной) системы и ее компонентов (ЗИС)
- реагирование на компьютерные инциденты (ИНЦ)
- управление конфигурацией (УКФ)
- управление обновлениями программного обеспечения (ОПО)
- планирование мероприятий по обеспечению безопасности (ПЛН)
- обеспечение действий в нештатных ситуациях (ДНС)
- информирование и обучение персонала (ИПО)

Для разнообразия можно ознакомиться с положением дел в Беларуси. Там есть вот такие приколные законы⁷:

- Об информатизации
- О научно-технической информации;
- О национальном архивном фонде и архивах в Республике Беларусь
- О печати и других средствах массовой информации
- О правовой охране программ для ЭВМ и баз данных
- О введении в действие Единой системы классификации и кодирования технико-экономической и социальной информации Республики Беларусь
- и др.

1.0.4 Защита от несанкционированного доступа

Когда мы начинаем говорить о безопасности бд, в первую очередь в голову приходит мысль, что не плохо было бы подумать о конфиденциальности (а уже потом про доступность и тд.). С нее и начнем.

К основным средствам защиты информации относят следующие⁸:

- идентификация и аутентификации
- установление прав доступа к объектам БД
- защита полей и записей таблиц БД
- шифрование данных и программ

Простейший пример аутентификации это парольная защита. Парольная защита представляет простой и эффективный способ защиты БД от несанкционированного доступа. Пароли устанавливаются конечными пользователями или администраторами БД и хранятся в определенных системных файлах СУБД в зашифрованном виде. Улучшенным вариантом является использование механизма SSL-аутентификации с использованием сертификатов.

В целях контроля использования основных ресурсов СУБД во многих системах имеются средства установления прав доступа к объектам БД. Права доступа определяют возможные действия над объектами. Владелец объекта, а также администратор БД имеют все права. Остальные пользователи к разным объектам могут иметь различные уровни доступа.

К данным, имеющимся в таблице, могут применяться меры защиты по отношению к отдельным полям и отдельным записям. В реляционных СУБД отдельные записи специально не защищаются. Применительно к защите данных в полях таблицы можно выделить такие уровни прав доступа, как полный запрет доступа, только чтение, разрешение всех операций (просмотр, ввод новых значений,

⁷А это взято с [лекций по администрированию баз данных](#)

⁸За оригиналом [сюда](#)

удаление, изменение). Более мощным средством защиты данных является их шифрование. Для расшифровки информации пользователи, имеющие санкционированный доступ к зашифрованным данным, имеют ключ и алгоритм расшифровки.

Итак, для минимизации риска несанкционированного доступа необходима реализация комплекса нормативных, организационных и технических защитных мер, в первую очередь: введение ролевого управления доступа, организация доступа пользователей по предъявлению сертификата, выборочное шифрование для сегментов базы данных.

1.0.5 Защита от вывода

Мы уже говорили об этом в угрозах безопасности 1.0.2 на 7 странице. У (Утебов Данияр Рашидович 2008) есть только объяснение логического вывода на основе функциональных зависимостей или ограничений целостности, а за решением нас посылают в (Смирнов 2007). О! Я тут неожиданно понял что все, что я находил в интернете, в статьях и других книгах это копияста различной степени наглости из (Смирнов 2007). Так что не буду отличаться оригинальностью и я. Помимо очевидного мониторинга с целью свести количество таких связей к минимуму, надо очень внимательно следить за привилегиями (принцип минимальных привилегий).

Привилегия – это разрешение на выполнение в системе определенного действия. Не имея соответствующей привилегии, пользователь не может получить доступ к данным или выполнить какое-либо действие.

Все привилегии могут быть разделены на два класса: системные привилегии и привилегии доступа к объектам.

Системная привилегия – это привилегия, которая дает пользователю право на выполнение какой-либо операции в масштабе базы данных. Например, пользователь с системной привилегией ALTER TABLESPACE может изменять любую табличную область (за исключением некоторых ограничений на табличную область SYSTEM).

Привилегия доступа к объекту – это разрешение пользователю на выполнение определенной операции над определенным объектом, например выполнение выборки из некоторой таблицы. При этом пользователь может формировать любые запросы к данной таблице, но не имеет права модифицировать данные этой таблицы или формировать какой-либо запрос к другой таблице.

1.0.6 Целостность БД

Когда мы говорили о защите от НСД, речь шла о конфиденциальности. Теперь настало время целостности.

По (Смирнов 2007):

Задача обеспечения целостности предусматривает комплекс мер по предотвращению непреднамеренного изменения или уничтожения информации, используемой информационной системой управления или системой поддержки принятия решений. Изменение или уничтожение данных может быть следствием неблагопри-

ятного стечения обстоятельств и состояния внешней среды (стихийные бедствия, пожары и т. п.), неадекватных действий пользователей (ошибки при вводе данных, ошибки операторов и т. п.) и проблем, возникающих при многопользовательской обработке данных.

Угроза нарушения целостности включает в себя любое умышленное или случайное изменение информации, обрабатываемой в информационной системе или вводимой из первичного источника данных. К нарушению целостности данных может привести как преднамеренное деструктивное действие некоторого лица, изменяющего данные для достижения собственных целей, так и случайная ошибка программного или аппаратного обеспечения, приведшая к безвозвратному разрушению данных.

Мы уже обсуждали основные угрозы целостности в 1.0.2 на 8.

Средства обеспечения целостности баз данных включают автоматическую поддержку некоторой системы правил, описывающих допустимость и достоверность хранимых и вводимых значений. Реляционная модель включает некоторые характерные правила, вытекающие из существа модели: ограничения домена и ограничения таблицы.

- Целостность домена предполагает, что допустимое множество значений каждого атрибута является формально определенным. То есть существуют формальные способы проверки того, что конкретное значение атрибута в базе данных является допустимым. Строка не будет вставлена в таблицу, пока каждое из значений ее столбцов не будет находиться в соответствующем домене (множестве допустимых значений).
- Целостность таблицы означает, что каждая строка в таблице должна быть уникальной. Хотя не все СУБД промышленного уровня требуют выполнения такого ограничения, возможность уникальной идентификации каждой строки представляется необходимой для большинства реальных приложений.

Ограничения целостности позволяют гарантировать, что требования к данным будут соблюдаться независимо от способа их загрузки или изменения. В большинстве СУБД предусмотрена поддержка следующих типов статических ограничений целостности⁹:

1. ограничение на определенность значения атрибута (NOT NULL)
2. ограничение на уникальность значения атрибутов (UNIQUE)
3. ограничение – первичный ключ
4. ограничение – внешний ключ
5. ограничение целостности, задаваемое предикатом

1.0.7 Аудит

Мало того, что все это великолепие надо долго и скрупулезно настраивать, так еще и за всем этим надо следить. О том, как, мы, опять же, находим у (Смирнов 2007).

⁹Немного другая точка зрения: <https://studfiles.net/preview/6354061/page:55/> (Роскомнадзор будет ругаться, но вас это пугать не должно)

Важнейшей составляющей процесса обеспечения безопасности ИС является проведение квалифицированного аудита. Проведение профессионального независимого аудита позволяет своевременно выявить существующие недостатки в системе обеспечения безопасности ИС и объективно оценить соответствие параметров, характеризующих режим обеспечения информационной безопасности, требуемому решаемым задачам организации уровню.

Полноценная система обеспечения безопасности ИС должна обладать развитыми средствами аудита, то есть, как минимум, СУБД должна обладать средствами автоматического ведения протоколов действий пользователей системы.

В СУБД средство ведения аудита может быть реализовано в виде независимой утилиты (IBM DB2) или возможностей, управляемых языковыми средствами системы (Oracle). Средства аудита обычно ассоциированы с экземпляром, т. е. для каждого экземпляра сервера баз данных может быть запущен собственный файл аудита.

Средства аудита выполняют фиксацию информации об активности пользователей системы в словаре данных или специальном файле – журнале аудита. Информация о настройках системы аудита хранится в специальном конфигурационном файле. Файл настройки или параметры команды активизации аудита определяют перечень событий, которые фиксируются системой аудита.

Пользователь, обладающий необходимыми полномочиями, может выполнять следующие действия со средствами аудита:

- запускать и останавливать средства аудита;
- просматривать состояние конфигурации средства аудита и настраивать средства аудита на фиксацию определенных событий;
- переписывать данные аудита во внешние файлы операционной системы для проведения независимого анализа.

1.0.8 Многоуровневая защита

Мы уже успели поговорить про уровни в связке интерфейс-СУБД-ОП-среда [1.0.1], сеть-ОС-БД [1.0.2] и даже ФСТЭКовские уровни [1.0.3].

Самое полное описание, пожалуй, сеть-ОС-БД, так что дальше будем отталкиваться от него. Вполне очевидно что защищать все и сразу сложно, каждый уровень имеет свою специфику, но какие то общие принципы выделить можно. Так как все труды на эту тему каскадно ссылаются на (Смирнов 2007), то просто процитирую:

Анализ наиболее успешных решений в области обеспечения информационной безопасности баз данных позволил сформулировать несколько полезных принципов, которыми можно руководствоваться при проектировании систем защиты:

- экономическая оправданность механизмов защиты
- открытое проектирование
- распределение полномочий между различными субъектами в соответствии с правилами организации

- минимально возможные привилегии для пользователей и администраторов
- управляемость системы при возникновении отказов и сбоев
- психологическая приемлемость работы средств защиты данных

1.0.9 Типы контроля безопасности: потоковый, контроль вывода, контроль доступа

Про все три типа мы уже успели поговорить. В частности 1.0.2 и 1.0.3. Единственное, что осталось уточнить – это контроль доступа. Идейно он этот вопрос распадается на два: *кому* и *что* мы будем разрешать?

Кому? Получение доступа к ресурсам информационной системы предусматривает выполнение трех процедур: идентификации, аутентификации и авторизации.

Общепринято, что 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 моделям, существующим в форме имен. Под аутентификацией понимается проверка и подтверждение подлинности образа идентифицированного субъекта, объекта, процесса. Как вытекает из самой сути данных понятий, в основе технологий идентификации и аутентификации лежит идеология вероятностного распознавания образов, обуславливая, соответственно, принципиальное наличие ошибок первого и второго рода.

1. Сущность процедуры идентификации состоит в назначении пользователю, т. е. объекту – потребителю ресурсов сервера баз данных – имени. Имя пользователя – это некоторая уникальная метка, соответствующая принятым соглашениям именования и обеспечивающая однозначную идентификацию объекта реального мира в пространстве отображаемых объектов. С позиций ИС источники, предъявившие идентификатор, неразличимы.
2. Сущность процедуры аутентификации состоит в подтверждении подлинности пользователя, представившего идентификатор.
3. Сущность процедуры авторизации состоит в определении перечня конкретных информационных ресурсов, с которыми аутентифицированному пользователю разрешена работа.

Процедуры идентификации, аутентификации и авторизации являются обязательными для любой защищенной ИС.

Что? В целом используют одну из трех моделей безопасности: дискреционная, мандатная и ролевая.

1. 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).

2. Мандатная модель доступа характерна для случая, когда возможность конкретных действий с данными или документами определяется внешним, обычно глобальным собственником информации. Исторически в роли такого глобального собственника чаще всего выступало государство. Основная идея мандатной модели доступа состоит в приписывании объектам и субъектам доступа меток. Если метки объекта и субъекта соответствуют (в некотором смысле), то субъект получает право на выполнение определенных действий с объектом. В роли метки, приписываемой субъектам в государственных структурах, выступает, например, форма допуска. В реляционной модели в качестве структуры, обладающей меткой, естественно выбрать кортеж.
3. В основу ролевой модели положена идея принадлежности всех данных системы некоторой организации, а не пользователю, как в случае моделей дискреционного и мандатного доступа. В целом модель ориентирована на упрощение и обеспечение формальной ясности в технологии обеспечения политики безопасности системы. Управление доступом в ролевой модели осуществляется как на основе матрицы прав доступа для ролей, так и с помощью правил, регламентирующих назначение ролей пользователям и их активацию во время сеансов.

В ролевой модели классическое понятие субъект разделяется на две части: 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 привилегий, необходимых для осуществления определенной деятельности. (Самым распространенным примером роли является присутствующая почти в каждой системе учетная запись администратора (например, root для UNIX и Administrator для Windows), который обладает специальными полномочиями и может использоваться несколькими пользователями.

Ролевая модель включает три компонента: модель отображения пользователь – роль, модель отображения привилегия – роль и модель отображения роль – роль. Для упрощения логической структуры объектов управления вводится понятие иерархии ролей. Роль, входящая в иерархию, может включать другие роли, наследуя все привилегии включаемых ролей.

2 Механизмы, поддерживающие высокую готовность

Определение: Высокая готовность (High Availability, HA) - это характеристика технической системы, разработанная для избежания невыполненного обслуживания путём уменьшения или управления сбоями и минимизации времени плановых простоев.

Высокая готовность для каждой конкретной системы зависит от цели, для которой предназначена эта система. Для некоторых компаний высокая готовность значит максимальное downtime за год в несколько минут. Для других HA может значит downtime несколько часов в месяц. Доступность (готовность) часто измеряется в «Nines»

| Availability % | Downtime per year ^[note 1] | Downtime per month | Downtime per week | Downtime per day |
|--------------------------------|---------------------------------------|--------------------|-------------------|------------------|
| 90% ("one nine") | 36.53 days | 73.05 hours | 16.80 hours | 2.40 hours |
| 95% ("one and a half nines") | 18.26 days | 36.53 hours | 8.40 hours | 1.20 hours |
| 97% | 10.96 days | 21.92 hours | 5.04 hours | 43.20 minutes |
| 98% | 7.31 days | 14.61 hours | 3.36 hours | 28.80 minutes |
| 99% ("two nines") | 3.65 days | 7.31 hours | 1.68 hours | 14.40 minutes |
| 99.5% ("two and a half nines") | 1.83 days | 3.65 hours | 50.40 minutes | 7.20 minutes |

Рис. 1: Пример Nines. Взято из википедии

HA включает в себя несколько важных аспектов:

1. Доступность данных
2. Защита данных
3. Производительность
4. Цена поддержки

2.1 Средства, поддерживающие высокую готовность

Аппаратная и программная поддержки Существует несколько возможных уровней обеспечения высокой готовности системы: аппаратный и программный уровни.

Аппаратный уровень включает в себя:

1. Репликация БД Репликация позволяет скопировать данные с одного сервера бд на другой. Существуют два подхода репликации баз данных, рассмотрим каждый из них:

- Репликация Master-Slave

В этом подходе выделяется один основной сервер базы данных, который называется Мастером. На нем происходят все изменения в данных (любые запросы INSERT/UPDATE/DELETE). Слейв сервер постоянно копирует все изменения с Мастера. С приложения на Слейв сервер могут отправляться запросы на чтение данных (запросы SELECT). Таким образом Мастер сервер может отвечать за изменения данных, а Слейв за чтение. Также Мастер сервер может отвечать за все операции с данными, а Слейв являться бэкапом. При выходе из строя Слейва, достаточно просто переключить все приложение на работу с Мастером. После этого восстановить репликацию на Слейве и снова его запустить. Если выходит из строя Мастер, нужно переключить все операции (и чтения и записи) на Слейв. Таким

образом он станет новым Мастером. После восстановления старого Мастера, настроить на нем реплику, и он станет новым Слейвом. Рассмотрена асинхронная репликация. При синхронной репликации результат сразу же записывается и в Мастер и в Слейв. Возвращение управления клиенту происходит только после записи в обе базы.

- Репликация Master-Master

В этой схеме, любой из серверов может использоваться как для чтения так и для записи. При использовании такого типа репликации достаточно выбирать случайное соединение из доступных Мастеров. Вероятные поломки делают Master-Master репликацию непривлекательной. Выход из строя одного из серверов практически всегда приводит к потере каких-то данных. Последующее восстановление также сильно затрудняется необходимостью ручного анализа данных, которые успели либо не успели скопироваться.

Как уже говорилось ранее, репликацию данных можно разделить на синхронную и асинхронную. При асинхронной репликации изменения в slave или 2 master могут появляться с задержкой. При синхронной репликации может существовать только одна версия данных, что накладывает ограничения на работу с данными в процессе репликации.

Существует логическая репликация. Логическая репликация — это метод репликации объектов данных и изменений в них, использующий репликационные идентификаторы (обычно это первичный ключ). Мы называем такую репликацию «логической», в отличие от физической, которая построена на точных адресах блоков и побайтовом копировании. PostgreSQL поддерживает оба механизма одновременно; Логическая репликация позволяет более детально управлять репликацией данных и аспектами безопасности.

В логической репликации используется модель публикаций/подписок с одним или несколькими подписчиками, которые подписываются на одну или несколько публикаций на публикующем узле. Подписчики получают данные из публикаций, на которые они подписаны, и могут затем повторно опубликовать данные для организации каскадной репликации или более сложных конфигураций.

Логическая репликация таблицы обычно начинается с создания снимка данных в публикуемой базе данных и копирования её подписчику. После этого изменения на стороне публикации передаются подписчику в реальном времени, когда они происходят. Подписчик применяет изменения в том же порядке, что и узел публикации, так что для публикаций в рамках одной подписки гарантируется транзакционная целостность. Этот метод репликации данных иногда называется транзакционной репликацией.

2. RAID - технология виртуализации данных для объединения нескольких физических дисковых устройств в логический модуль для повышения отказоустойчивости и производительности. Рассмотрим базовые уровни рейд массивов:

- (a) RAID 0 (striping — «чередование») — дисковый массив из двух или более жёстких дисков без резервирования.
- (b) RAID 1 (mirroring — «зеркалирование») — массив из двух (или более) дисков, являющихся

полными копиями друг друга. Не следует путать с массивами RAID 1+0 (RAID 10), RAID 0+1 (RAID 01), в которых используются более сложные механизмы зеркалирования.

- (с) RAID 2. Массивы такого типа основаны на использовании кода Хэмминга. Диски делятся на две группы: для данных и для кодов коррекции ошибок, причём если данные хранятся на $2^n - n - 1$ дисках, то для хранения кодов коррекции необходимо n дисков. Суммарное количество дисков при этом будет равняться $2^n - 1$. Данные распределяются по дискам, предназначенным для хранения информации, так же, как и в RAID 0, то есть они разбиваются на небольшие блоки по числу дисков.
- (d) В массиве RAID 3 из n дисков данные разбиваются на куски размером меньше сектора (разбиваются на байты или блоки) и распределяются по $n - 1$ дискам. Ещё один диск используется для хранения блоков чётности. В RAID 2 для этой цели применялся $n - 1$ диск, но большая часть информации на контрольных дисках использовалась для коррекции ошибок «на лету», в то же время большинство пользователей устраивает простое восстановление информации в случае её повреждения, для чего хватает данных, уместяющихся на одном выделенном жёстком диске.

Отличия RAID 3 от RAID 2: невозможность коррекции ошибок на лету.

- (e) RAID 4 похож на RAID 3, но отличается от него тем, что данные разбиваются на блоки, а не на байты. Таким образом, удалось отчасти «победить» проблему низкой скорости передачи данных небольшого объёма. Запись же производится медленно из-за того, что чётность для блока генерируется при записи и записывается на единственный диск.
- (f) RAID 5. Основным недостатком уровней RAID от 2-го до 4-го является невозможность производить параллельные операции записи, так как для хранения информации о чётности используется отдельный контрольный диск. RAID 5 не имеет этого недостатка. Блоки данных и контрольные суммы циклически записываются на все диски массива, нет асимметричности конфигурации дисков. Под контрольными суммами подразумевается результат операции XOR (исключающее или). Хор обладает особенностью, которая даёт возможность заменить любой операнд результатом, и, применив алгоритм хор, получить в результате недостающий операнд.
- (g) RAID 6 — похож на RAID 5, но имеет более высокую степень надёжности — два (или более) диска данных и два диска контроля чётности. Основан на кодах Рида — Соломона и обеспечивает работоспособность после одновременного выхода из строя любых двух дисков. Обычно использование RAID 6 вызывает примерно 10-15 % падение производительности дисковой группы, относительно RAID 5, что вызвано большим объёмом работы для контроллера (более сложный алгоритм расчёта контрольных сумм), а также необходимостью читать и перезаписывать больше дисковых блоков при записи каждого блока.

Также существуют различные комбинации данных подходов.

Программные подходы:

1. Автоматический перезапуск экземпляров БД, сетевых демонов и других ресурсов
2. Защита от повреждения данных(Data corruption protection)

- Использование контрольных чек-сумм
 - Автоматическое восстановление из резервной копии или повторная передача
3. PITR (Point In Time Recovery) Данный механизм позволяет восстановить базу данных в том виде, в котором она была в каком-то моменте в прошлом.
 4. Application continuity (Oracle) Маскирует от приложения и конечных пользователей падение базы данных и позволяет восстановить сеансы базы данных во время работы.
 5. WAL Данный подход будет описан ниже.

Кластерная организация серверов баз данных

Кластеризация базы данных - это процесс объединения нескольких серверов или инстансов, соединяющих одну базу данных. Иногда одного сервера может быть недостаточно для управления объемом данных или количеством запросов, тогда возникает необходимость в кластере.

К общим требованиям, предъявляемым к кластерным системам, относятся:

1. Высокая готовность
2. Высокое быстродействие
3. Масштабирование
4. Удобство обслуживания

Кластеры баз данных являются распространенной технологией. Рассмотрим три типа архитектуры кластерных вычислений. Отказоустойчивые кластеры, высокопроизводительные кластеры и кластеры балансировки нагрузки.

1. Отказоустойчивые / высокодоступные кластеры. Кластер обеспечивает доступность сервиса путем репликации серверов и избыточной реконфигурации программного и аппаратного обеспечения. Таким образом, каждая система контролирует другую и работает на запросы, если какой-либо один из узлов выходит из строя.
2. Высокопроизводительные кластеры. Целью разработки высокопроизводительных кластеров баз данных является создание высокопроизводительных компьютерных систем. Основная цель - разумное распределение рабочей нагрузки.
3. Кластеры балансировки нагрузки. Эти кластеры базы данных служат для распределения нагрузки между различными серверами. Они стремятся обеспечить увеличенную пропускную способность сети, в конечном итоге увеличивая производительность. Системы в этой сети объединяют свои узлы, с помощью которых пользовательские запросы равномерно распределяются между участвующими узлами.

Несмотря на всю распределенную систему на заднем плане, пользователю это кажется единой системой. Использование кластеров варьируется от предприятия к предприятию, в зависимости от вида процессов и требуемого уровня производительности.

Параметры настройки СУБД

Непонятно о чем это, настройки чего

Сохранение и восстановление БД

Основным свойством транзакций СУБД является durability. Идея механизма обеспечения этого свойства является одинаковой для всех СУБД. СУБД использует специальный лог (Write-Ahead Lock - WAL), в который записываются все транзакции, результат которых не попал на физический диск. WAL — это журнал опережающей записи, технология, которую часто используют для улучшения надежности базы данных. Когда вы заносите информацию в базу данных, база выполняет несколько операций, манипулируя блоками. Следовательно, с точки зрения файловой системы INSERT и UPDATE не являются атомарными операциями: если кто-то внезапно выключит ваш сервер, данные окажутся испорчены. Если возникает сбой, база данных использует этот файл для того, чтобы восстановить данные на момент падения. WAL – это бинарный лог, так что для его чтения нужна специальная утилита. Например в PostgreSQL данный файл называется WAL и лежит по пути `$PGDATA/pg_xlog`. PostgreSQL позволяет отключать WAL для отдельных таблиц, помечая их как UNLOGGED.

2.2 Оперативное администрирование

Задачи, средства и режимы администрирования Задачи администратора баз данных могут незначительно отличаться в зависимости от вида применяемой СУБД, но в основные задачи входит:

- Проектирование базы данных.
- Оптимизация производительности базы данных.
- Резервирование и восстановление базы данных.
- Обеспечение целостности баз данных.
- Обеспечение перехода на новую версию СУБД.

Существуют различные программы и способы администрирования БД, которые напрямую зависят от БД. Основным инструментом работы с базой данных является DSL и программы, например для Oracle можно использовать SQL Developer. Существуют программы, поддерживающие работу с любой БД, например продукты JetBrains. Также отдельно существуют средства мониторинга серверов.

Мониторинг серверов СУБД Мониторинг СУБД можно условно разделить на два типа:

1. Мониторинг средствами СУБД
2. Мониторинг хостов, на которых запущена БД.

Мониторинг средствами СУБД

Средства мониторинга, предоставляемые СУБД зависят в первую очередь от конкретной реализации этой системы, однако можно выделить метрики, которые в большинстве случаев собираются встроенным мониторингом. В первую очередь это объем операций ввода/вывода, необходимых

для исполнения транзакции, утилизация процессоров и временем отклика системы. Наиболее распространенной метрикой оценки производительности системы является ее время отклика, которое представляет собой интервал времени, в течении которого сервер возвращает первую строку результата исполнения запроса, т.е. пользователь получает визуальное подтверждение того, что его запрос исполняется. Пропускная способность обслуживаемых сервером процессов и пользователей определяет сколько запросов возможно исполнить в фиксированный интервал времени, и сколько строк и какого размера возвращается клиенту. При увеличении числа активных процессов и/или пользователей, возрастает и их конкуренция за системные ресурсы. Результатом такой чрезмерной нагрузки может стать увеличение времени отклика и снижение общей пропускной способности. Большое влияние на производительности базы данных оказывает также физическая и логическая целостность данных.

Мониторинг хостов, на которых запущена БД.

Многофункциональным средством является Zabbix — свободная система мониторинга и отслеживания статусов разнообразных сервисов компьютерной сети, серверов и сетевого оборудования. Он поддерживает несколько видов мониторинга. Simple checks — может проверять доступность и реакцию стандартных сервисов, таких как SMTP или HTTP без установки какого-либо программного обеспечения на наблюдаемом хосте. Zabbix agent — может быть установлен на UNIX-подобных или Windows хостах для получения данных о нагрузке процессора, использовании сети, дисковом пространстве и так далее. External check — выполнение внешних программ. Zabbix также поддерживает мониторинг через SNMP. Благодаря расширяемости данная система мониторинга позволяет контролировать любые метрики системы. К недостаткам можно отнести некоторую трудоемкость при необходимости слежения за нестандартными параметрами системы. Для мониторинга систем чаще всего используется протокол SNMP, и можно сказать, что он являлся долгое время стандартом де-факто. Протокол SNMP работает на базе протокола UDP и предназначен для использования сетевыми управляющими станциями. Он позволяет управляющим станциям собирать информацию, изменять, посылать «trap» на сервер мониторинга. Протокол определяет формат данных, их обработка и интерпретация остаются на усмотрение системного администратора и системы мониторинга. SNMP-сообщения не имеют фиксированного формата и фиксированных полей. Как следствие протокол SNMP универсален, но его главным недостатком является дополнительный трафик и необходимость загружать резидентные продукты (агенты) на управляемых объектах.

2.3 Функциональная насыщенность СУБД

Формы избыточности Системы, обеспечивающие непрерывный доступ к данным (fault tolerant) или почти непрерывный (high availability) обычно опираются на различные формы избыточности. Как правило, это системы дублирования аппаратного обеспечения и контролируемой избыточности данных.

Аппаратная избыточность Аппаратная избыточность может включать платформы с полным резервированием, поддерживающие (standby) процессоры, диски с двойным интерфейсом (dual-port), дисковые массивы и пр. Один из вариантов - зеркалирование дисков, когда один диск используется в качестве копии другого и может быть использован при сбое вместо него. Хотя аппаратная избыточность и важна для повышения общей надежности системы, ее реализация, как правило, не ориентирована на обработку транзакций СУБД и на связанные с этим специфические

ограничения, например, обеспечение атомарности транзакции. В результате СУБД не может воспользоваться преимуществами чисто аппаратных решений резервирования системы для повышения своей производительности.

Избыточность данных Нормальная форма — свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, потенциально приводящей к логически ошибочным результатам выборки или изменения данных. Нормальная форма определяется как совокупность требований, которым должно удовлетворять отношение.

Процесс преобразования отношений базы данных к виду, отвечающему нормальным формам, называется нормализацией. Нормализация предназначена для приведения структуры БД к виду, обеспечивающему минимальную логическую избыточность, и не имеет целью уменьшение или увеличение производительности работы или же уменьшение или увеличение физического объёма базы данных. Конечной целью нормализации является уменьшение потенциальной противоречивости хранимой в базе данных информации. Как отмечает К. Дейт, общее назначение процесса нормализации заключается в следующем:

- исключение некоторых типов избыточности;
- устранение некоторых аномалий обновления;
- разработка проекта базы данных, который является достаточно «качественным» представлением реального мира, интуитивно понятен и может служить хорошей основой для последующего расширения;
- упрощение процедуры применения необходимых ограничений целостности.

Устранение избыточности производится, как правило, за счёт декомпозиции отношений таким образом, чтобы в каждом отношении хранились только первичные факты (то есть факты, не выводимые из других хранимых фактов).

Существуют 6 нормальных форм.

Программное зеркалирование Программное зеркалирование дисков, называемое также дуплексированием (duplexing) или мультиплексированием (multiplexing), может не только защитить от аппаратных сбоев, но и улучшить производительность. Поскольку зеркалирование базы данных (или ее частей - таблиц(ы), индексов, их фрагментов и пр.) производится на другом физическом устройстве, то операции чтения данных можно распределить между двумя устройствами и производить параллельно (рис. 2). Конечно, зеркалирование бесполезно с любой точки зрения, если оно организовано на одном диске. В случае повреждения зеркалируемого диска все операции автоматически переносятся на исправный диск, сбойный диск выводится в отключенное состояние, причем приложения не замечают каких-либо изменений в конфигурации системы. После замены неисправного диска параллельно с работой пользователей запускается процесс оперативной синхронизации зеркальных дисков (on-line mirroring), на физическом уровне копирующий рабочий диск.

Тиражирование данных Тиражирование в системах, требующих в первую очередь повышенной надежности, в целом подобно зеркалированию, но здесь копия данных может поддерживаться удаленно. Если происходит копирование всей базы данных, то обычно это делается с целью обеспечить горячий резерв (warm standby). Однако в некоторых реализациях есть возможность использовать копию для просмотра (без модификации) данных. Это способно обеспечить значительные преимущества для систем со смешанной загрузкой, поскольку приложения для принятия решений, генерации отчетов и т.п. могут обращаться к копии базы данных, в то время как приложения оперативной обработки транзакций используют первичную базу данных.

2.4 Системы, обладающие свойством высокой готовности

Описание, назначение, примеры

Cassandra - распределённая система управления базами данных, относящаяся к классу NoSQL-систем и рассчитанная на создание высокомасштабируемых и надёжных хранилищ огромных массивов данных, представленных в виде хэша.

Хранилище само позаботится о проблемах наличия единой точки отказа (single point of failure), отказа серверов и о распределении данных между узлами кластера (cluster node). При чем, как в случае размещения серверов в одном центре обработки данных (data center), так и в конфигурации со многими центрами обработки данных, разделенных расстояниями и, соответственно, сетевыми задержками. Под надёжностью понимается итоговая согласованность (eventual consistency) данных с возможностью установки уровня согласования данных (tune consistency) каждого запроса.

Узлы кластера кассандры равноценны, и клиенты могут соединиться с любым из них, как для записи, так и для чтения. Запросы проходят стадию координации, во время которой, выяснив при помощи ключа и разметчика на каких узлах должны располагаться данные, сервер посылает запросы к этим узлам. Будем называть узел, который выполняет координацию — координатором (coordinator), а узлы, которые выбраны для сохранения записи с данным ключом — узлами-реплик (replica nodes). Физически координатором может быть один из узлов-реплик — это зависит только от ключа, разметчика и меток. Для каждого запроса, как на чтение, так и на запись, есть возможность задать уровень согласованности данных.

Когда данные приходят после координации на узел непосредственно для записи, то они попадают в две структуры данных: в таблицу в памяти (memtable) и в журнал закрепления (commit log). Таблица в памяти существует для каждого колоночного семейства и позволяет запомнить значение моментально. Технически это хеш-таблица (hashmap) с возможностью одновременного доступа (concurrent access) на основе структуры данных, называемой “списками с пропусками” (skip list). Журнал закрепления один на всё пространство ключей и сохраняется на диске. Журнал представляет собой последовательность операций модификации. Так же он разбивается на части при достижении определённого размера. Такая организация позволяет сделать скорость записи ограниченной скоростью последовательной записи на жесткий диск и при этом гарантировать долговечность данных (data durability). Журнал закрепления в случае аварийного останова узла читается при старте сервиса кассандры и восстанавливает все таблицы в памяти. Получается, что скорость упирается во время последовательной записи на диск, а у современных жёстких дисков это порядка 100МБ/с. По этой причине журнал закрепления советуют вынести на отдельный дисковый носитель.

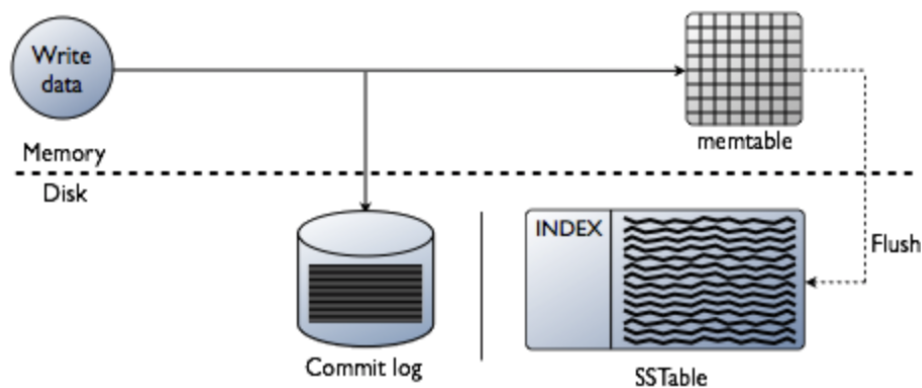


Рис. 2: Схема записи в Cassandra

3 Защита данных в распределенных системах

3.1 Распределенные вычислительные среды

Распределённая обработка данных - методика выполнения прикладных программ группой систем. При этом пользователь получает возможность работать с сетевыми службами и прикладными процессами, расположенными в нескольких взаимосвязанных абонентских системах. (Т.Ю.Сергеева, М.Ю.Сергеев 2014)

Распределенная обработка информации в среде клиент-сервер. Концепция распределенной вычислительной среды Distributed Computing Environment (DCE). Распределенные базы данных в сетях ЭВМ

Компьютер (или программу), управляющий ресурсом, называют сервером этого ресурса (файл-сервер, сервер базы данных, вычислительный сервер...). Клиент и сервер какого-либо ресурса могут находиться как в рамках одной вычислительной системы, так и на различных компьютерах, связанных сетью. Основной принцип технологии "клиент-сервер" заключается в разделении функций приложения на три группы:

- ввод и отображение данных (взаимодействие с пользователем);
- прикладные функции, характерные для данной предметной области;
- функции управления ресурсами (файловой системой, базой данных и т.д.)

Поэтому, в любом приложении можно выделить следующие компоненты:

- компонент представления данных
- прикладной компонент
- компонент управления ресурсом

Связь между компонентами осуществляется по определенным правилам, которые называют "протокол взаимодействия". Каждый из компонентов приложения при этом может работать на

выделенном сервере (узле) или разделять ресурсы сервера с другими компонентами приложения. В связи с этим можно выделить следующие модели приложений:

- двухзвенная модель (модель «клиент-сервер»)
- трехзвенная модель (модель сервера приложений)
- многозвенная модель

Двухзвенная модель позволяет распределить различным образом три компонента приложения между двумя узлами. **Трехзвенная модель** предполагает выделение для каждого из трех компонентов приложения свой сервер. **Многозвенная модель** позволяет отдельным компонентам использовать ресурсы нескольких серверов, например, распределенные базы данных. Компанией Gartner Group, специализирующейся в области исследования информационных технологий, предложена следующая классификация двухзвенных моделей взаимодействия клиент-сервер (Бомонка 2019)

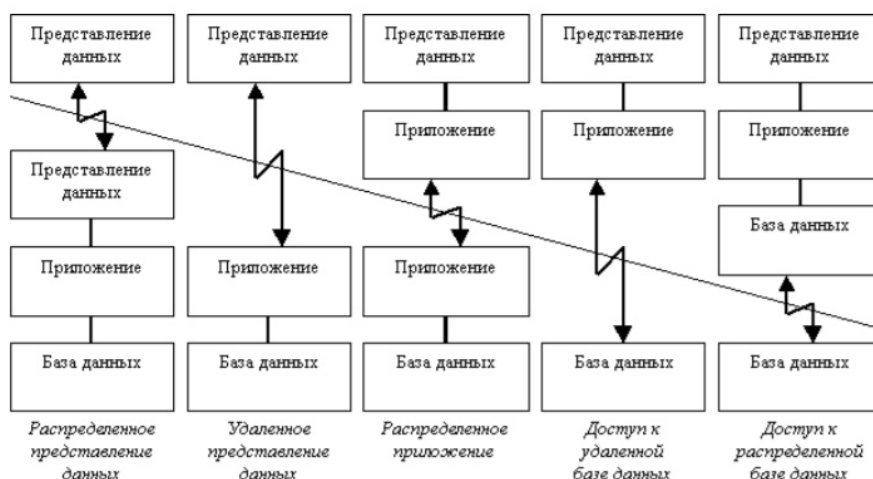


Рис. 3: Классификация двухзвенных моделей

3.2 Угрозы безопасности распределенных СУБД

Угрозы доступности, целостности и конфиденциальности данных. Механизмы противодействия

В данном материале (В. Ю. Статъев, В.А. Тиньков 2020) предпринята попытка изложить комплексный подход к обеспечению безопасности распределенных информационных систем (ИС). Однако в рамках одной статьи не представляется возможным подробно описать все аспекты информационной безопасности, поэтому поднятые вопросы рассмотрены очень кратко. Однако были приложены все усилия к тому, чтобы не упустить ни одну из проблем так или иначе влияющую на безопасность ИС.

При рассмотрении информационной безопасности распределенных ИС выбран клиент/серверный подход, ввиду того, что он позволяет провести декомпозицию ИС на отдельные компоненты, обеспечивающие определенные сервисы, а также потому, что ряд услуг по безопасности может быть реализован в виде отдельных серверов.

Кроме этого необходимо отметить, что обеспечение абсолютной защищенности информации в распределенных ИС - задача, практически, не достижимая, что обусловлено не только техническими вопросами, но и человеческим фактором. Поэтому вопрос может стоять только об определенном уровне защищенности.

Современный подход к информационной безопасности

Под информационной безопасностью понимается защита информации и поддерживающей ее инфраструктуры с помощью совокупности программных, аппаратно-программных средств и методов, а также организационных мер, с целью недопущения причинения вреда владельцам этой информации или поддерживающей ее инфраструктуре.

При построении системы защиты должен учитываться комплексный подход в обеспечении безопасности информации. Он подразумевает использование защитных механизмов на всех этапах жизненного цикла системы, от ее проектирования и до вывода из эксплуатации, и совместное решение целого спектра вопросов, начиная от физической защиты объектов ИС, с применением интеллектуальной системы контроля доступа, и заканчивая вопросами поддержки нормального функционирования ИС в критических ситуациях.

Проектирование системы безопасности информации (СБИ) осуществляется совместно с проектированием самой информационной системы. При внесении любых изменений в структуру ИС, это должно найти адекватное отражение и в системе защиты.

При разработке систем информационной безопасности необходимо учитывать передовые тенденции развития информационных технологий, к каким в настоящее время относятся интрасети (Интранет - Intranet). Характерными чертами данных сетей является то, что они основываются на технологии клиент/сервер, имеют в своем составе разнородные корпоративные информационные системы и пользуются внешними сервисами, базовым для которых является протокол TCP/IP, а также предоставляет аналогичные сервисы вовне. Центральным элементом интрасетей является WEB-сервис, поэтому чрезвычайно важным является вопрос обеспечения защищенности этого сервиса.

Разработка политики безопасности

Первым шагом на пути построения СБИ является разработка политики безопасности ИС. Под политикой безопасности следует понимать утвержденный высшим должностным лицом организации, в интересах которой разрабатывается ИС и система защиты, документ, с указанием организационных и технических мероприятий, направленных на защиту информации, и поддерживающей ее инфраструктуры.

В данном документе должны найти обязательное отражение основные принципы безопасности информации, такие как многоуровневая защита и разнообразие средств и методов защиты, минимизация привилегий, разделение полномочий, невозможность обхода средств защиты и т.п.

Построение модели нарушителя

После разработки политики безопасности производится анализ угроз ИС, возможных каналов утечки информации и построение на этой основе модели потенциального нарушителя.

Согласно предлагаемой модели, в качестве нарушителя может выступать как отдельное физическое лицо, так и специальные программы, содержащие в себе некие деструктивные элементы - разрушающие программные компоненты (РПК).

Под объектом защиты в данной модели понимается материальный носитель защищаемой информации, а также сама информация, доступ к которым определен режимом разграничения доступа, реализуемым системой защиты, а также поддерживающая инфраструктура.

Предполагается, что по отношению к ИС нарушитель может быть двух типов: внутренним или внешним. При этом внутренний нарушитель выступает в качестве законного пользователя системы или лица из числа обслуживающего персонала или администрации ИС, а внешний является лицом, не подпадающим ни под одну из указанных категорий. РПК, в свою очередь, может быть отнесено к категории внутреннего нарушителя.

Возможные угрозы нарушителя напрямую зависят от его типа. Возможности внутреннего нарушителя ограничены установленными правилами разграничения доступа, внутриобъектовым режимом и его функциональными обязанностями. Возможности РПК в ИС намного шире и определяются работоспособностью ПО, в котором они заложены, а также наличием в системе средств активного аудита и мониторинга.

Проектирование СБИ

Следующим шагом после разработки модели нарушителя является проектирование СБИ. Согласно принципу многоуровневой защиты и разнообразия средств и методов защиты, можно выделить следующие самостоятельные системы в составе СБИ (при этом необходимо учитывать, что все они интегрированы и находятся в тесной взаимосвязи):

- Система контроля доступа на объекты и в помещения ИС
- Система защиты информации в ИС от НСД
- Средства защиты от РПК
- Средства поддержания доступности информации в ИС

Система контроля доступа

Система контроля доступа на объекты и в помещения, в которых функционирует ИС, должна состоять из трех подсистем: пропускная подсистемы (ПП), подсистемы охранной сигнализации (ПОС) и подсистемы телевизионного наблюдения (ПТН).

Простейшим вариантом пропускной подсистемы являются одиночные кодовые замки, устанавливаемые в защищаемых помещениях, либо электронные замки с карточками доступа - ЭЗКД. Учитывая, что в ЭЗКД предусмотрен режим постановки на охрану, то ПП может быть интегрирована с ПОС.

Что касается ПОС и ПТН, то они не отделимы одна от другой и функционируют под управлением единого вычислительного комплекса.

Система защиты информации в ИС от НСД

Данная система может быть реализована как встроенными средствами защиты сетевых операционных систем и систем управления базами данных (СУБД), так и дополнительными средствами. Стоит лишь отметить, что при построении системы защиты от НСД необходимо, по возможности, избегать дублирования многих функций, с тем, чтобы СБИ не страдала избыточностью, что в конечном итоге скажется как на удобстве работы пользователей и их желании выполнять все предписания системы защиты, так и на управляемости самой СБИ.

В состав указанной системы также должны входить:

- средства контроля, управления и идентификации при удаленном доступе
- средства управления доступом и идентификации в рамках ИС
- средства экранирования ИС от открытых сетей, а также разноуровневых сетей внутри данной ИС друг от друга
- средства управления, анализа и аудита СБИ в рамках сетевых конфигураций

Средства контроля, управления и идентификации при удаленном доступе к ИС

Указанная категория средств предназначена для осуществления процедуры контроля подключения к ИС удаленных пользователей, а также для управления их доступом и осуществления идентификации. Данная проблема возникает в связи со структурой интрасетей, когда удаленные пользователи получают доступ к ресурсам корпоративной ИС по выделенным, либо коммутируемым каналам связи. В этом случае существует реальная вероятность несанкционированного подключения нарушителя к линии связи, с активным или пассивным прослушиванием сети и выдачей себя за зарегистрированного пользователя ИС. В целях недопущения подобного целесообразно применение методов с использованием одноразовых паролей и единого входа в сеть.

Принцип систем с одноразовыми паролями основан на однократном использовании пароля для процедуры идентификации в ИС, в результате чего перехват его нарушителем становится бессмысленным. Идентификация и аутентификация пользователя осуществляется сервером безопасности (аутентификации), входящим в состав ИС.

Концепция единого входа в сеть предоставляет существенное удобство для пользователей, так как при подключении к ИС им достаточно только один раз доказать свою подлинность. Кроме этого, применение этой концепции способствует усилению информационной безопасности, ввиду того, что в сети отсутствует открытая передача аутентификационной информации.

Средства управления доступом и идентификации в рамках ИС

Учитывая территориальную разнесенность современных ИС и использование технологии клиент/сервер, целесообразно выделение функции проверки подлинности в виде отдельного сервера безопасности (аутентификации). Услугами данного сервера должны пользоваться все другие серверы и пользователи ИС.

Сервер безопасности может быть реализован в виде одного или нескольких серверов, функционирующих на физически защищенных компьютерах. Серверы должны содержать репозиторий субъектов ИС и их секретные ключи.

При построении сервера безопасности возможны два пути:

- Сервер безопасности с центральным хранением данных репозитория
- Сервер безопасности с децентрализованным хранением данных репозитория

Однако централизованное хранение является нецелесообразным по следующим причинам:

- сильная зависимость от готовности сервера
- возможность расширения или комбинирования сетей в процессе развития ИС

Дополнительной причиной децентрализации сервера безопасности является объединение сетей при развитии ИС. Возможна ситуация, когда две до того независимые сети, имеющие каждая в своем составе отдельный сервер безопасности, объединяются. В этом случае один из серверов должен быть уничтожен, а все определенные в нем данные по авторизации переданы на другой сервер. Если же связь между двумя сетями устанавливается временно, то необходимо наличие нескольких серверов безопасности. В противном случае, один сервер должен будет производить постоянные, двойные определения то для одной, то для другой сети. Для того, чтобы избежать этого, необходимо наличие нескольких серверов безопасности.

Средства экранирования ИС

Средства экранирования используются для подключения ИС к открытым сетям или развязки разноуровневых сетей, т.е. сетей, обрабатывающих информацию с различным грифом секретности. Концепция систем типа Firewall (брандмауер, межсетевой экран) была разработана для снижения риска нелегального доступа к закрытой информации при подключении частных сетей (в том числе ЛВС) к сетям общего пользования. Межсетевой экран представляет собой программно-аппаратный комплекс, размещенный на стыке двух сетей и реализующий следующие три функции:

- обеспечение обмена данными между сетями только через указанную систему
- фильтрация трафика обмена
- предотвращение возможности проникновения в сам экран

В этом случае обеспечивается эффективная блокировка внешнего трафика частной сети и жесткий контроль за ним. Кроме того экраны могут осуществлять разграничение доступа между различными сегментами одной корпоративной сети, а также контроль за информационными потоками, направленными во вне, обеспечивая тем самым необходимый режим конфиденциальности.

Применение экранов также позволяет существенно уменьшить уязвимость внутренних сервисов безопасности, так как нарушителю необходимо вначале преодолеть защитные механизмы самого экрана, где они сконфигурированы особенно тщательно.

Существующие в настоящее время экраны могут быть условно разделены на следующие четыре типа:

- экраны с фильтрацией пакетов (packet-filtering firewall)
- плюзы сеансового уровня (circuit-level gateway)
- плюзы прикладного уровня (application-level gateway)
- экраны экспертного уровня (stateful inspection firewall)

Однако лишь некоторые экраны могут быть отнесены только к одной из указанных категорий. При этом необходимо отметить, что экраны экспертного уровня обеспечивают один из самых высоких на сегодняшний день уровней безопасности интрасетей.

Средства управления, анализа и аудита

Средства аудита занимают свое особое положение в ряду средств обеспечения безопасности информации, заключающееся в том, что все действия нарушителя по преодолению средств защиты фиксируются, позволяя тем самым вовремя обнаружить попытку несанкционированного входа в ИС. Причем, учитывая принцип многоуровневой защиты, нарушителю придется преодолевать несколько защитных рубежей, что будет обязательно отмечено в регистрационном протоколе. В случае если указанная процедура выполняется в режиме реального времени, администратором безопасности могут быть своевременно предприняты соответствующие меры по предотвращению незаконного вторжения на одном из следующих уровнях защиты.

Кроме средств аудита часть программных продуктов также позволяет осуществлять оценку системы безопасности сети, имитируя все известные способы, применяемые нарушителями для проникновения в интрасети, и тем самым, обнаруживая в системе защиты слабые места. Данные программные продукты не только выявляют уязвимые места, но и определяют действия, которые необходимо предпринять для ликвидации пробелов в сетевой системе безопасности. Администратору остается лишь выбрать способы их устранения.

Средства защиты от РПК

Учитывая, что среди отнесенных к РПК компонентов особое распространение получили компьютерные вирусы, в данном разделе описывается подход по осуществлению антивирусной защиты в ИС.

Так как сети, построенных по технологии Интранет, являются разнородными системами, важную роль приобретает использование антивирусных средств (АВС), изначально ориентированных на работу в сети и поддерживающих многоплатформность. В связи с этим особую актуальность приобретает использование АВС, предназначенных для работы в смешанных сетях NetWare и Windows NT. Наряду с этим, учитывая, что ключевым сервисом интрасетей является WEB-сервис, возникает необходимость применения антивирусного сканера, предназначенного для осуществления проверки наличия вирусов в файлах, загружаемых программами просмотра WEB-страниц (WEB-браузерами), получаемых с FTP-серверов и в почтовых сообщениях, действуя совершенно прозрачно для пользователей. Активизация АВС происходит автоматически при запуске WEB-браузера.

Особенности организации защиты информации от компьютерных вирусов

Система защиты от вирусов должна строиться исходя из обязательного выполнения следующих процедур:

- входной контроль новых программных средств и входной контроль поступающей по сети информации и данных (выполняется применяемыми АВС)
- защиту операционной системы и системных программ от заражения (создание АВС типа ViruSaf e Multi-LAN специальной аварийной дискеты с COMMAND.COM, IPX и NETX)
- карантинный режим эксплуатации нового программного обеспечения

- резервирование главной загрузочной записи (MBR), таблицы размещения файлов (FAT) и CMOS-памяти (применение ABC ViruSafe Multi-LAN, IncoluLAN и т.п.)

Средства резервного копирования

Резервное копирование программ и данных необходимо проводить с целью минимизации потерь в случае отказов оборудования, либо сбоев в программном обеспечении ИС. Данная задача наиболее сложна именно в интрасетях с их распределенными ресурсами и неоднородностью, в которых работают компьютеры под управлением различных операционных систем. Учитывая клиент/серверный характер интрасетей функцию резервного копирования целесообразно также выделить в виде отдельного сервера (сервера архива).

Распространение клиент/серверного подхода на процедуру резервного копирования информации и данных имеет ряд преимуществ по сравнению с традиционными методами. Они выражаются в следующем:

- Администраторы рабочих групп освобождаются от необходимости согласования действий и самой процедуры создания локальных резервных копий
- Единообразие процедуры создания резервных копий в ИС
- Возможность мониторинга процесса резервирования и диагностики возникших проблем

Одним из способов обеспечения высокой доступности информации является создание резервных копий с возможностью ее хранения в двух местах: один экземпляр хранится поблизости от оригинала, а другой в удаленном безопасном месте.

Безопасность систем управления базами данных

Составной частью информационной безопасности ИС является безопасность систем управления базами данных (СУБД). Учитывая, что СУБД является ключевым элементом современной ИС, можно отметить, что для них важны все три аспекта информационной безопасности: конфиденциальность, целостность и доступность.

В СУБД для идентификации и проверки подлинности применяются либо соответствующие механизмы операционной системы, либо специальный SQL-оператор.

Обеспечение конфиденциальности данных

В СУБД, как правило, используется произвольное управление доступом, когда владелец объекта передает права доступа к нему (привилегии) по своему усмотрению. При этом привилегии в СУБД можно подразделить на две категории: привилегии безопасности и привилегии доступа.

Привилегии безопасности всегда выделяются конкретному пользователю и позволяют выполнять административные действия.

Привилегии доступа определяют права доступа субъектов к определенным объектам.

Специфическим механизмом управления доступом в СУБД являются представления. Они позволяют сделать видимыми для субъектов только те столбцы базовых таблиц, доступ к которым предоставлен субъектам администратором базы.

Поддержание целостности

Целостность данных не менее важна, чем конфиденциальность, ввиду того, что для баз данных, как и для ИС в целом, главными врагами являются не внешние нарушители, а ошибки оборудования, программ, администраторов и пользователей системы.

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

Ограничения могут относиться как к таблицам, так и к отдельным столбцам. Они накладываются владельцами таблицы и оказывают влияние на все операции с данными.

Правила позволяют вызывать выполнение заданных действий при определенных изменениях базы данных. В отличие от ограничений, являющихся лишь средствами контроля простых условий, правила позволяют создавать сколь угодно сложные соотношения между различными элементами базы данных.

Обеспечение доступности данных

Доступность данных подразумевает обеспечение информационной системы средствами поддержания высокой доступности. Поддержание высокой доступности позволяет свести к минимуму возможные сбои аппаратного обеспечения, в частности носителей информации, а также ошибки обслуживающего персонала и программного обеспечения. В качестве мер поддержания высокой доступности может быть названа кластеризация сервера баз данных (выделение нескольких компьютеров, выполняющих общее приложение), а также тиражирование данных (хранение базы данных в различных местах).

Угрозы СУБД

СУБД отличаются от других компонентов ИС специфичными угрозами, и главным их источником является сама природа баз данных. Известно, что основным средством общения с СУБД выступает язык SQL, являющийся мощным инструментом манипулирования данными. С его помощью, используя механизм правил, могут быть созданы сложные, трудно поддающиеся анализу цепочки действий, позволяющие не явным образом передавать право на выполнение определенных процедур тем, кто не имеет на это полномочий.

В качестве примера можно привести несколько угроз, возникающих при использовании языка SQL: получение информации путем логических выводов, агрегирование данных, покушение на высокую доступность.

Методы борьбы против получения информации путем логических выводов состоят в тщательном проектировании модели данных, иерархии привилегий и видимых пользователям представлений.

Агрегирование данных состоит в получении новой информации путем комбинирования данных, полученным официальным путем. Причем информация, содержащаяся в скомбинированных данных, может иметь гриф более высокий, чем первичная информация.

Методом борьбы с агрегированием может быть тщательное проектирование модели данных и максимально допустимое ограничение доступа пользователей к информации.

Покушение на высокую доступность может быть реализовано, если пользователю-нарушителю доступны все возможности языка SQL. При этом он легко сможет заблокировать работу других пользователей. Поэтому, в целях борьбы с данным видом угроз, рекомендуется запрещать непосредственный SQL-доступ к базе данных, используя для этого серверы приложений.

3.3 Распределенная обработка данных

Понятие распределенной транзакции

Если данные хранятся в одной базе данных, то транзакция к ней рассматривается как локальная. В распределенных базах транзакция, выполнение которой заключается в обновлении данных на нескольких узлах сети, называется глобальной или **распределенной транзакцией**.

Внешне выполнение распределенной транзакции выглядит как обработка транзакции к локальной базе данных. Тем не менее распределенная транзакция включает в себя несколько локальных транзакций, каждая из которых завершается двумя путями — фиксируется или прерывается. Распределенная транзакция фиксируется только в том случае, когда зафиксированы все локальные транзакции, ее составляющие.

Модель обработки транзакций

В стандарте ANSI/ISO SQL определены модель транзакций и функции операторов COMMIT и ROLLBACK. Стандарт определяет, что транзакция начинается с первого SQL-оператора, инициируемого пользователем или содержащегося в программе. Все последующие SQL-операторы составляют тело транзакции. Транзакция завершается одним из четырех возможных путей:

- оператор COMMIT означает успешное завершение транзакции; его использование делает постоянными изменения, внесенные в базу данных в рамках текущей транзакции
- оператор ROLLBACK прерывает транзакцию, отменя изменения, сделанные в базе данных в рамках этой транзакции; новая транзакция начинается непосредственно после использования ROLLBACK
- успешное завершение программы, в которой была инициирована текущая транзакция, означает успешное завершение транзакции (как будто был использован оператор COMMIT)
- ошибочное завершение программы прерывает транзакцию (как будто был использован оператор ROLLBACK)

Точки сохранения применяются, как правило, в протяженных транзакциях и позволяют разделить транзакцию на несколько небольших осмысленных фрагментов. Пользователь может зафиксировать работу в любой точке транзакции с тем, чтобы выполнить ее откат к состоянию, соответствующему этой точке.

Откат и фиксация транзакций становятся возможными благодаря журналу транзакций. Он используется следующим образом. Известно, что все операции над реляционной базой данных суть операции над строками таблиц. Следовательно, для обеспечения отката таблиц к предыдущим состояниям достаточно хранить не состояния таблицы, а лишь те ее строки, которые подверглись изменениям.

При выполнении любого оператора SQL, который вносит изменения в базу данных, СУБД автоматически заносит очередную запись в журнал транзакций. Запись состоит из двух компонентов: первый - это состояние строки до внесения изменений, второй - ее же состояние после внесения изменений. Только после записи в журнал транзакций СУБД действительно вносит изменения в базу данных. Если после данного оператора SQL был выполнен оператор COMMIT, то в журнале транзакций делается отметка о завершении текущей транзакции. Если же после оператора SQL

следовал оператор ROLLBACK, то СУБД просматривает журнал транзакций и отыскивает записи, отражающие состояние измененных строк до внесения изменений. Используя их, СУБД восстанавливает те строки в таблицах базы данных, которые были изменены текущей транзакцией, - таким образом аннулируются все изменения в базе данных.

Мониторы обработки транзакций

Мониторы обработки транзакций (Transaction Processing Monitor - TPM), или, проще, мониторы транзакций - программные системы (которые относят к категории middleware, то есть к посредническому или промежуточному программному обеспечению), обеспечивающие эффективное управление информационно-вычислительными ресурсами в распределенной системе. Они представляют собой гибкую, открытую среду для разработки и управления мобильными приложениями, ориентированными на оперативную обработку распределенных транзакций. В числе важнейших характеристик TPM - масштабируемость, поддержка функциональной полноты и целостности приложений, достижение максимальной производительности обработки данных при невысоких стоимостных показателях, поддержка целостности данных в гетерогенной среде. TPM опираются на трехзвенную модель "клиент-сервер" (модель сервера приложений или AS-модель), описанную в Разделе 2. Естественно, что все преимущества модели отражаются и на программных системах, построенных на ее основе.

На современном рынке мониторов транзакций основными "действующими лицами" являются такие системы, как ACMS (DEC), CICS (IBM), TOP END (NCR), PATHWAY (Tandem), ENCINA (Transarc), TUXEDO System (USL). Несмотря на принципиальное сходство, конкретные TPM отличаются рядом характеристик, причем различия часто вытекают как из специфики операционной системы, в которой реализован и функционирует TPM.

Корпоративная среда обработки транзакций

TPM на базе UNIX опирается на фундаментальное понятие - корпоративную среду обработки транзакций (Enterprise Transaction Processing - ETP). Архитектура ETP - это три ряда компьютеров:

- Ряд 1: Персональные станции (Personal Workstations);
- Ряд 2: Компьютеры под управлением ОС UNIX (UNIX Transaction Processing Servers - UPTS);
- Ряд 3: Mainframe-системы (Proprietary Transaction Processing Servers - PTPS) или компьютеры под управлением UNIX с RISC-архитектурой процессоров;

Компьютеры **ряда 1**, функционирующие под управлением DOS, MS Windows, OS/2, UNIX, используются в качестве рабочих мест конечных пользователей. Характерная черта ETP - отсутствие ограничений на модели компьютеров, составляющих этот ряд. Однако, как правило, ряд 1 состоит из компьютеров на базе процессоров Intel 486/Pentium под управлением MS Windows (MS Windows фактически стала стандартом оконного графического интерфейса для большинства категорий пользователей и стандартом операционной среды для подавляющего числа прикладных программ и систем).

Ряд 2 составляют компьютеры среднего класса под управлением ОС UNIX, на которых функционирует ядро TPM и, как правило, реляционные СУБД (Oracle, Informix, Ingres), выступающие

в качестве менеджера ресурсов. Кроме того, на них же может быть установлен шлюз к ТРМ в операционной среде мэйнфрейма (как правило, разработчики ТРМ на базе UNIX предусматривают в конфигурации своих систем шлюз к наиболее популярной такой системе - IBM CICS).

Ряд 3 представлен мэйнфреймами или RISC-компьютерами под управлением UNIX. О мэйнфреймах мы говорим в тех ситуациях, когда исторически сложилось так, что в организации они существуют уже долгое время, берут на себя большую часть всего объема обработки транзакций, концентрируют огромные вычислительные ресурсы и содержат большие массивы данных (то есть речь идет об унаследованных системах). Если этого "тяжелого наследия" нет, то можно смело использовать в качестве компьютеров ряда 3 RISC-серверы, сегодня приближающиеся по производительности к мэйнфреймам.

Таким образом, среда обработки транзакций формируется из набора разнородных компьютеров (и соответствующих ОС), ранжируемых от персональных компьютеров до мэйнфрейм-систем. ТРМ на базе UNIX представляет собой своего рода "клей который связывает вместе компьютеры трех рядов в открытую унифицированную среду обработки транзакций.

Ключом к интеграции систем, функционирующих на компьютерах различных рядов, является специализированный интерфейс прикладного программирования ATMI (Application Transaction Manager Interface), обеспечивающий:

- для ряда 1 - формирование и передачу запросов от клиентов к серверам, выполняющимся на компьютерах ряда 2
- для ряда 2 - обработку запросов, поступающих от компьютера ряда 1 (в том числе и с обращением к менеджеру ресурсов), и, по необходимости, формирование и направление запросов к серверам, выполняющимся на компьютерах ряда 3
- для ряда 3 - обработку запросов, поступающих от серверов ряда 2

Стоит отметить, что подобное представление о корпоративной среде обработки транзакций не является абстракцией. Сегодня многие организации приходят именно к такой, "трехуровневой" архитектуре информационных систем (с той оговоркой, что наличие ряда 3 вызвано историческими причинами - мэйнфреймы использовались первоначально и сразу от них отказаться невозможно).

3.4 Протоколы фиксации

Протоколы фиксации

В локальной системе базы данных для принятия транзакции диспетчер транзакций должен только передать решение о фиксации диспетчеру восстановления. Однако в распределенной системе диспетчер транзакций должен передать решение о фиксации всем серверам в различных сайтах, где выполняется транзакция, и обеспечить единообразное выполнение решения. Когда обработка завершается на каждом сайте, она достигает состояния частично подтвержденной транзакции и ожидает, пока все другие транзакции достигнут их частично подтвержденных состояний. Когда он получает сообщение о том, что все сайты готовы к фиксации, он начинает фиксировать. В распределенной системе либо все сайты фиксируют, либо ни один из них не делает.

Различные протоколы распределенной фиксации:

- Однофазный коммит
- Двухфазный коммит
- Трехфазный коммит

Распределенная однофазная фиксация

Распределенная однофазная фиксация – это самый простой протокол фиксации. Давайте рассмотрим, что есть контролирующий сайт и несколько подчиненных сайтов, где выполняется транзакция. Шаги в распределенном коммите:

- После того, как каждое ведомое устройство локально завершило свою транзакцию, оно отправляет сообщение «ГОТОВО» на контролирующий сайт
- Подчиненные ожидают сообщения «Подтвердить» или «Прервать» с контролирующего сайта. Это время ожидания называется окном уязвимости
- Когда контролирующий сайт получает сообщение «СДЕЛАНО» от каждого ведомого, он принимает решение о фиксации или отмене. Это называется точкой фиксации. Затем он отправляет это сообщение всем рабам
- При получении этого сообщения ведомое устройство либо фиксирует, либо прерывает работу, а затем отправляет подтверждающее сообщение на контролирующий сайт

После того, как каждое ведомое устройство локально завершило свою транзакцию, оно отправляет сообщение «ГОТОВО» на контролирующий сайт.

Подчиненные ожидают сообщения «Подтвердить» или «Прервать» с контролирующего сайта. Это время ожидания называется окном уязвимости .

Когда контролирующий сайт получает сообщение «СДЕЛАНО» от каждого ведомого, он принимает решение о фиксации или отмене. Это называется точкой фиксации. Затем он отправляет это сообщение всем рабам.

При получении этого сообщения ведомое устройство либо фиксирует, либо прерывает работу, а затем отправляет подтверждающее сообщение на контролирующий сайт.

Распределенная двухфазная фиксация Распределенная двухфазная фиксация снижает уязвимость однофазных протоколов фиксации. Шаги, выполняемые на двух этапах:

Этап 1

- После того, как каждое ведомое устройство локально завершило свою транзакцию, оно отправляет сообщение «ГОТОВО» на контролирующий сайт. Когда контролирующий сайт получил сообщение «ГОТОВО» от всех подчиненных, он отправляет сообщение «Подготовить» подчиненным
- Рабы голосуют за то, хотят ли они по-прежнему совершать или нет. Если ведомый хочет зафиксировать, он отправляет сообщение «Готово»
- Раб, который не хочет коммитить, отправляет сообщение «Не готов». Это может произойти, если ведомое устройство имеет конфликтующие параллельные транзакции или истекло время ожидания

После того, как каждое ведомое устройство локально завершило свою транзакцию, оно отправляет сообщение «ГОТОВО» на контролирующий сайт. Когда контролирующий сайт получил сообщение «ГОТОВО» от всех подчиненных, он отправляет сообщение «Подготовить» подчиненным.

Рабы голосуют за то, хотят ли они по-прежнему совершать или нет. Если ведомый хочет зафиксировать, он отправляет сообщение «Готово».

Раб, который не хочет коммитить, отправляет сообщение «Не готов». Это может произойти, если ведомое устройство имеет конфликтующие параллельные транзакции или истекло время ожидания.

Этап 2

- После того, как контролирующий сайт получил сообщение «Готово» от всех ведомых –
 - Контролирующий сайт отправляет сообщение «Global Commit» подчиненным
 - Подчиненные устройства применяют транзакцию и отправляют сообщение «Подтвердить АСК» на контролирующий сайт
 - Когда контролирующий сайт получает сообщение «Подтвердить АСК» от всех ведомых устройств, он считает транзакцию подтвержденной
- После того, как контролирующий сайт получил первое сообщение «Не готов» от любого ведомого –
 - Контролирующий сайт отправляет сообщение «Global Abort» подчиненным
 - Слэйвы отменяют транзакцию и отправляют сообщение «Abort АСК» на контролирующий сайт
 - Когда контролирующий сайт получает сообщение «Abort АСК» от всех ведомых устройств, он считает транзакцию отмененной

Распределенная трехфазная фиксация Шаги в распределенной трехфазной фиксации следующие:

- Этап 1: Шаги такие же, как при распределенной двухфазной фиксации
- Этап 2: Подготовьтесь к фиксации
 - Контролирующий сайт выдает широковещательное сообщение «Ввод подготовленного состояния»
 - Ведомые сайты голосуют «ОК» в ответ
- Фиксация / Отмена

Этапы аналогичны двухфазной фиксации, за исключением того, что сообщение «Подтвердить АСК» / «Прервать АСК» не требуется

Защищенные протоколы фиксации

Обработка распределенных транзакций в базах данных с многоуровневой секретностью (MLS)

Известно, что в MLS/DBMS не ко всем данным, содержащимся в базе данных, доступ осуществляется одинаково. Однако современные СУБД, как правило, не имеют адекватных средств диагностики и механизма определения того, что пользователь имеет возможность доступа только к тем данным, которые являются релевантными. Таким образом, MLS/DBMS отличается от соответствующих DBMS, по крайней мере, следующими двумя особенностями:

- каждый элемент данных в базе данных связан с уровнем доступа
- доступ пользователя к данным должен контролироваться релевантностью для данного пользователя

Разработка сервиса MLS/DBMS в современных компьютерных системах представляет много проблем. До настоящего времени внедрение многоуровневого разграничения доступа в операционную систему представляет собой значительные трудности. Решение этой проблемы в виде аббревиатуры обозначается TCB. Хотя в разрешении вопросов TCB для удаленных пользователей в MLS/DBMS вводятся компромиссы, остается много проблем, которые требуется разрешать. Наиболее очевидная проблема состоит в том, что вопросы классификации в СУБД значительно сложнее, чем в файловых системах и могут быть сложнее реализованы. Другая проблема состоит в том, что для классификации данных, содержащих контекстные представления, временные параметры, их композицию, необходимы унифицированные базы данных.

3.5 Тиражирование данных

Тиражирование данных - это асинхронный перенос изменений объектов исходной базы данных (source database) в базы данных, принадлежащие к различным узлам распределенной системы. Функции тиражирования данных возложены на специальный компонент СУБД - сервер тиражирования данных, называемый репликатором (replicator), задача которого состоит в обеспечении идентичности данных в принимающих базах данных (target database) данным в исходной БД.

Обзор средств тиражирования данных

На практике установка тиражируемой системы сводится к чисто административным действиям, для выполнения которых необходимо получить ответы на четыре вопроса: **Что** тиражировать? **Где** тиражировать? **Когда** тиражировать? **Как разрешать** предполагаемые конфликты?

Что тиражировать? Как и любая другая методология, тиражирование имеет собственную терминологию. Краеугольным камнем тиражирования является понятие согласованного распределенного набора данных (consistent distributed data set - CDDS). Фактически, это тот самый набор данных в базе (или база данных целиком), идентичность которого на всех узлах, вовлеченных в процесс тиражирования, и поддерживает репликатор. Здесь и далее мы будем говорить об узлах распределенной системы, хотя участвовать в тиражировании могут и базы данных, расположенные на одном узле.

CDDS может быть представлен следующими конфигурациями данных:

- Вся база данных

- Избранные объекты базы данных: таблицы или представления
- Вертикальные проекции объектов БД: избранные столбцы таблиц и/или представлений
- Горизонтальные подмножества объектов: избранные записи из таблиц и/или представлений
- Сочетание наборов 2-4

Где тиражировать?

Следующий основополагающий элемент схемы тиражирования - это путь переноса изменений (data propagation path - DPP) из каждой тиражируемой базы данных в другие БД. Гибкость тиражирования в значительной степени обеспечивается богатством выбора способа передачи данных между узлами распределенной системы.

Практика эксплуатации распределенных систем подсказала следующие схемы тиражирования данных, реализуемые репликатором:

- Центр-филиалы изменения в БД филиалов переносятся в центральную БД, и /или наоборот
- Равноправное, несколько БД разделяют общий набор изменяемых и тиражируемых данных
- Каскадное, изменения в одной БД переносятся в другую БД, откуда в свою очередь в третью БД и т. д., эта схема позволяет оптимизировать баланс загрузки серверов баз данных, расположенных на различных узлах
- Через шлюзы изменения в базе данных могут переноситься в БД другой СУБД
- Различные комбинации всех перечисленных выше схем

Когда тиражировать?

После определения тиражируемого набора данных и маршрута переноса изменений остается определить лишь момент инициации репликационного сервера. Как уже говорилось раньше, элементарным изменением, вызывающим реакцию репликатора, является транзакция, но тиражировать каждую транзакцию по одиночке было бы не всегда удобным. Например, было бы трудно зафиксировать состояние в принимающей базе данных на определенный час. Стремясь быть максимально гибким, репликатор предоставляет следующие возможности:

- тиражирование начинается после завершения определенного числа транзакций, в том числе и после каждой транзакции
- тиражирование происходит через равные промежутки или к определенному моменту времени
- процесс тиражирования контролируется вручную администратором системы или созданным пользователем монитором тиражирования

Как разрешать конфликты?

Конфликты, возникающие в некоторых ситуациях, например, при встречном тиражировании или при восстановлении базы данных с помощью репликатора из реплицированной копии, можно отнести к разряду планируемых проблем. Репликатор при необходимости самостоятельно обнаруживает противоречия в тиражируемых данных и предоставляет разрешение конфликта администратору (противоречивые данные обязательно регистрируются в журнале), либо делает это автоматически. Возможны следующие варианты:

- разрешение конфликта в пользу более раннего или более позднего изменения
- разрешение конфликта в пользу наивысшего приоритета тиражируемой записи

Эффективные алгоритмы тиражирования

Алгоритмы тиражирования:

- Алгоритм восстановления данных при тиражировании
- Алгоритм создания копии при тиражировании данных

Более подробно об алгоритмах тиражирования можно почитать тут (https://www.hse.ru/data/2010/08/04/121Lyadova_Strelkov.pdf)

Сравнение подходов к тиражированию БД

3.6 Интеграция БД и Internet

Современные тенденции

Обзор существующих технологий

Вопросы безопасности: угрозы и методы противодействия

Перспективы развития

4 Безопасность в статистических БД

Инфы про статистические базы данных на русском языке мало и большая часть бредовая. Гуглите statistical database. Все источники будут на английском, переведено специально для вас. Источников несколько Lawrie Brown 2008 Дейт К. Дж. 2014, под конец была найдена статья, NAVIL R. ADAM 1989 в которой есть вся собранная инфа.

4.1 Общие сведения

. Определение статистической БД. Классификация статистических БД. Характеристики статистических БД.

Статистической (в приведенном здесь контексте) называется база данных, в которой допускаются запросы с обобщением данных (суммированием, вычислением среднего значения и т.д.), но не допускаются запросы по отношению к элементарным данным. Например, в статистической базе данных разрешается выдача запроса "Какова средняя зарплата программистов? тогда как выдача запроса "Какова зарплата программиста Мэри?"запрещена. Проблема статистических баз данных заключается в том, что иногда с помощью логических заключений на основе выполнения разрешенных запросов можно вывести ответ, который прямо может быть получен только с помощью

запрещенного запроса. "Обобщенные значения содержат следы исходной информации, и она может быть восстановлена злоумышленником после соответствующей обработки этих обобщенных значений. Такой процесс называется логическим выводом конфиденциальной информации". Практически для любой статистической базы данных всегда может быть определен общий трекер (в отличие от множества индивидуальных трекеров). Общий трекер (general tracker) — это логическое выражение, которое может быть использовано для поиска ответа на любой запрещенный запрос, т.е. запрос, включающий недопустимое логическое выражение. (В противоположность этому индивидуальный трекер работает только на основе запросов, включающих конкретные запрещенные выражения.) Требуется поддерживать баланс между репрезентативностью данных и конфиденциальностью отдельных записей.

4.2 Классификация

Классифицирует по следующим признакам:

- Чисто статистическая база данных - Обычная база данных со статистическим доступом

этот тип базы данных хранит только статистические данные. Примером может служить база данных переписи населения. этот тип базы данных содержит отдельные записи. Кроме того, база данных поддерживает набор статистических пользователей, которым разрешены только статистические запросы. Для этих последних пользователей совокупная статистика, основанная на базовых необработанных данных, генерируется в ответ на запрос пользователя или может быть предварительно вычислена и сохранена как часть базы данных.

- Оффлайн-Онлайн

В онлайн-SDB существует прямое взаимодействие пользователя с данными в режиме реального времени через терминал. В автономном SDB пользователь не контролирует обработку данных и не знает, когда выполняется его запрос данных. В этом режиме методы защиты, отслеживающие профили пользователей, становятся более громоздкими. Компромиссные методы, требующие большого количества запросов (например, метод компромисса на основе регрессии, см. Раздел 7), также усложняются при работе в автономном режиме.

- Статическая-Динамический

Статическая база данных-это такая база данных, которая никогда не меняется после ее создания. Большинство баз данных переписи являются статическими. Всякий раз, когда создается новая версия базы данных, эта новая версия считается другой статической базой данных. В отличие от этого, динамические базы данных могут изменяться непрерывно. Эта особенность может значительно усложнить проблему безопасности, поскольку частые выпуски новых версий могут позволить ищейкам использовать различия между версиями способами, которые трудно предвидеть. Методы возмущения данных могут не подходить для динамических СДБ, так как усилия по преобразованию исходного СДБ в возмущенный могут стать непомерными.

- Централизованный-Децентрализованный

В централизованном SDB есть одна база данных. В децентрализованном (распределенном) SDB перегруженные подмножества базы данных хранятся на различных узлах, Соединенных

коммуникационной сетью. Распределенная база данных может быть полностью реплицирована, частично реплицирована или секционирована. Проблема безопасности распределенного SDB является более сложной, чем проблема централизованного SDB из-за необходимости дублировать на каждом узле накладные расходы по контролю безопасности, а также трудности интеграции профилей пользователей. Выделенная-общая компьютерная система в выделенной SDB компьютерной системе используется исключительно для обслуживания приложений SDB. В общей системе приложения SDB работают на одной аппаратной системе с другими приложениями (возможно, с использованием различных баз данных). Общую среду защитить сложнее, так как другие приложения могут вмешиваться в защищаемые данные непосредственно через операционную систему, минуя механизм безопасности SDB.

4.3 Угрозы статистических БД

Об безопасности в статистических БД Деннинг [16.6]

"методы нарушения защиты данных просты и не связаны с большими расходами. Поэтому требование обеспечения полной секретности конфиденциальной информации несовместимо с требованием возможности вычисления точных статистических показателей для произвольных подмножеств данных в базе. По крайней мере одно из этих требований должно быть снято прежде, чем можно будет поверить в гарантии обеспечения секретности.

То есть существует мнение, что нельзя нормально обезопасить СБД.

Основная проблема СБД – проблема вывода. В общих чертах, проблема вывода для SDB может быть сформулирована следующим образом. Характеристическая функция C определяет подмножество записей (строк) в базе данных. Запрос, использующий C , предоставляет статистику по выбранному подмножеству. Если подмножество достаточно мало, возможно, даже одна запись, спрашивающий может быть в состоянии сделать вывод о характеристиках одного человека или небольшой группы. Даже для больших подмножеств характер или структура данных могут быть такими, что несанкционированная информация может быть выпущена. Для злоумышленника, который пытается достать индивидуальные данные задача состоит в том, чтобы сделать общий трекер. Нужно придумать такую последовательность запросов, чтобы вывести индивидуальную информацию. Суть в том, что можно набрать несколько запросов и проделать операции над ними там, чтобы деанонить записи.

Например для статичной СБД можно сделать такое

запрос1: выдать $a1 + a2 + a3$, запрос2: выдать $a1 + a2$. запрос1 - запрос2 = $a3$.

Для динамической онлайн СБД Мы хотим достать зарплату Мэри. Знаем, что ей 20 лет. Записываем в базу много левых записей с возрастом = 20 лет и нулевой зарплатой. Делаем запрос с минимальной агрегацией людей с возрастом 20 лет. Таким образом можно узнать возможную зп Мэри. Чем больше исходных знаний есть о Мэри, тем точнее получим данные. Также не стоит забывать что речь идет о БД. Поэтому статистическая БД наследует все опасности обычных БД.

4.4 Защита в статистических БД

Разделяют четыре общих подхода защиты статистических БД:

- концептуальный;
- ограничение запросов;
- возмущение данных;
- возмущение вывода;

На рисунке 4 изображена схема 3 подходов.

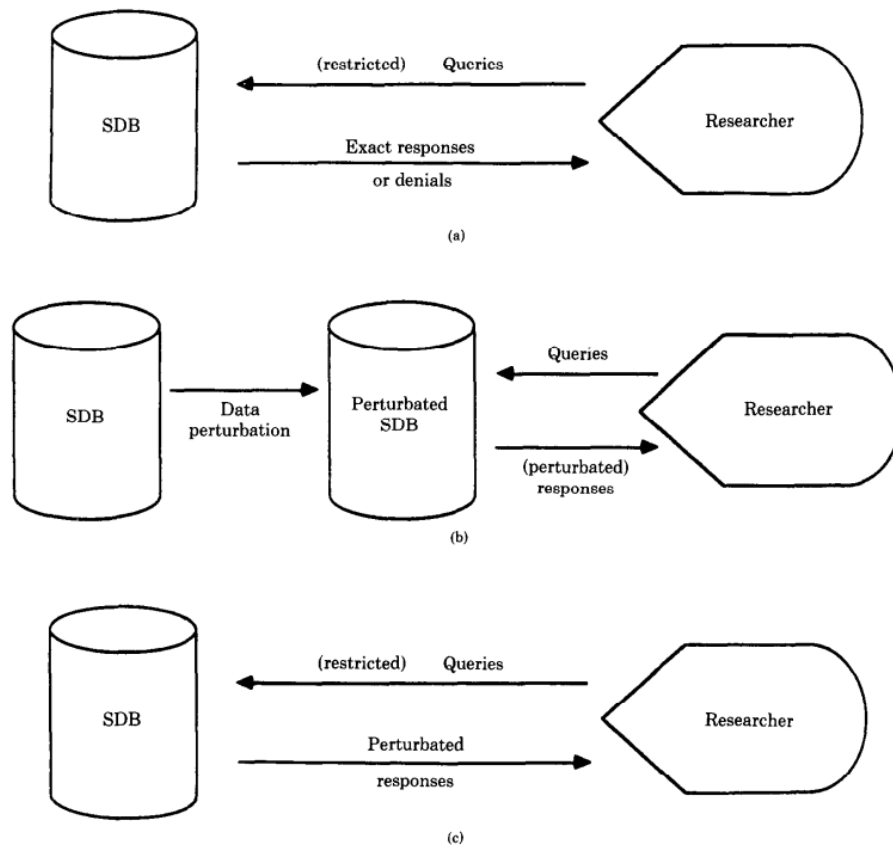


Рис. 4: Схемы безопасности СДБ

Концептуальный Основная проблема в том, что вся реляционная алгебра позволяет выводить индивидуальные данные. Концептуальный подход предлагает заменить классическую систему реляционной БД работающую с индивидуальными данными. Основной подход: Partitioning of the data base – Записи хранятся не отдельно, а агрегировано и обезличено. Судя по-всему похоже на microaggregation. Microaggregation – есть исходные данные, выделим несколько записей, посчитаем для них средние значения, заменим значения записей полученным средним, повторить для остальных записей в исходных данных

Еще один способ – решетчатая модель. Агрегация данных по разным признакам на разном уровне детализации.

Ну и естественно не забываем про классические способы. Access Restriction - пусть вы доктор, тогда вам можно всё. А исследователям и прочим анонам даём только некоторые записи или даже готовые стат.результаты по данным (агрегация)

Ограничение запросов Query Set Restriction – Ограничение на минимальный размер выдаваемой выборки.

Limiting intersection of query sets – Метод защиты блокирует такие запросы которые приводят к выводу данных через пересечение множеств запросов. Сохранение исторических сведений о результатах выполнения запросов и отклонение любых запросов, использующих значительное количество исходных данных, обработанных предыдущим запросом. Auditing - ведём логи, ловим негодяев и всяких подознительных

Возмущение данных Data Perturbation – добавим небольшой, случайный шум. Тогда точные данные вроде как и будут уничтожены, но можно заиграться и исказить репрезентативность

Случайное добавление дополнительных строк к основному набору данных, обрабатываемому запросом. Также предлагается организовать "обмен данными"("data swapping"), т.е. обмен значениями атрибутов между кортежами, осуществляемый таким образом, чтобы поддерживалась лишь статистическая точность. При этом даже если злоумышленнику удастся идентифицировать отдельное значение (например, некоторое значение зарплаты), то у него не будет способа узнать, какому именно кортежу (в нашем примере — сотруднику) оно принадлежит. Сложность этого подхода заключается в необходимости отыскать множество тех записей, между которыми можно будет организовать обмен значениями соответствующим образом. Подобные затруднения имеют место и при использовании большинства других методов.

Возмущение вывода Output Perturbation – как предыдущее, но преобразовывать будем результат каждого запроса Random Sampling – сделать так, чтобы один и тот же запрос давал каждый раз разные наборы записей Использование для ответов на запросы только случайных наборов исходных данных

Критерии безопасности Как было сказано гарантированной безопасности в СДБ не добиться. Поэтому делают оценку важности деанонимизированных данных и статистической точности (из-за наших всех действий для защиты данных)

Security – сферическая вероятность раскрыть (в том числе частично) запись в сбд Для каждого разрешенного агрегирующего запроса оценивается критерий безопасности. На основе количества инфы в базе можно подсчитать сколько тех или иных запросов нужно сделать для деанонимизации. На основе этого рассчитывается минимальная выборка для таких запросов.

Consistency – оценивается консистентность данных для методов возмущения данных. Оценивается степень возмущения данных. Создаем агрегирующий запрос и замеряем его изменения после внесенных возмущений. Robustness – оценка зависимости между возмущением и конкретной записью. Было бы круто чтобы возмущение не было зависимо от данных, но нужно каким-то образом сохранить ее статистические свойства.

Costs – Назначаем каждому запросу конкретную цену. Пользователю даем начальную сумму на которую он может делать запросы, такую чтобы он не смог вывести данные. Выглядит больше как

метод, а не критерий.

Список литературы

- NABIL R. ADAM, JOHN C. WORTMANN (1989). «Security-Control Methods for Statistical Databases: A Comparative Study». В: *ACM Computing Surveys*, Vol. 21, 4. URL: <https://dl.acm.org/doi/pdf/10.1145/76894.76895>.
- Смирнов (2007). *Безопасность систем баз данных*. Гелиос АРВ. ISBN: 9785854381635. URL: <https://www.twirpx.com/file/1706071/>.
- Lawrie Brown, William Stallings (2008). *Computer Security Principles and Practice*. Pearson. ISBN: 9780134794105.
- Утебов Данияр Рашидович, Белов Сергей Валерьевич (2008). «Классификация угроз в системах управления базами данных». В: *Вестник Астраханского государственного технического университета* 1, с. 87—92. URL: <https://cyberleninka.ru/article/n/klassifikatsiya-ugroz-v-sistemah-upravleniya-bazami-dannyh/viewer>.
- Лихоносов А. Г. (2011). *Интернет-курс по дисциплине Безопасность баз данных*. URL: http://www.e-biblio.ru/book/bib/01_informatika/b_baz_dan/sg.html (дата обр. 01.03.2020).
- Дейт К. Дж. (2014). *Введение в системы баз данных [7 издание]*. Уфимский Государственный Авиационный Технический Университет.
- Т.Ю.Сергеева, М.Ю.Сергеев (2014). *Распределенная обработка данных*. URL: https://cchgeu.ru/upload/iblock/cf5/metod_roi_ivt_ras_24.06.2016.pdf (дата обр. 20.04.2020).
- Бомонка (2019). *Распределенная обработка данных*. URL: https://ru.bmstu.wiki/%D0%A0%D0%B0%D1%81%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BE%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85 (дата обр. 20.04.2020).
- Мысев Алексей Эдуардович, Морозов Николай Владимирович (2019). «Правовое регулирование информационной безопасности в Российской Федерации». В: *Отечественная юриспруденция* 3, с. 51—55. URL: <https://cyberleninka.ru/article/n/pravovoe-regulirovanie-informatsionnoy-bezopasnosti-v-rossiyskoy-federatsii>.
- В. Ю. Статъев, В.А. Тиньков (2020). *Информационная безопасность распределенных информационных систем*. URL: <http://emag.iis.ru/arc/infosoc/emag.nsf/BPA/9191949adeb83654c32575be0039f17c> (дата обр. 20.04.2020).