РОССИЙСКАЯ ФЕДЕРАЦИЯ МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ ФГАОУ ВО ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК

КАФЕДРА ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Нестерова О. А.

АДМИНИСТРИРОВАНИЕ И БЕЗОПАСНОСТЬ MS SQL SERVER

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

ТЮМЕНЬ 2020

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Целью дисциплины «Администрирование и безопасность MS SQL Server» является формирование у студентов совокупности профессиональных качеств, обеспечивающих решение проблем, связанных с использованием и проектированием баз данных под управлением современных систем управления базами данных (СУБД), а также связанных с обеспечением безопасности информации в автоматизированных информационных системах (АИС), основу которых составляют базы данных (БД), навыкам работы со встроенными в системы управления базами данных (СУБД) средствами защиты

Задачей курса является:

- изучение моделей и механизмов защиты в СУБД;
- приобретение практических навыков организации защиты БД
- обучить студентов проводить обоснование и выбор рационального решения по защите систем управления баз данных с учетом заданных требований;

Изучение курса основано на следующих дисциплинах: «Разработка защищенных прикладных решений на базе современных систем управления базами данных», «Безопасность баз данных»

В результате изучения дисциплины студенты должны знать:

- основные информационные технологии, используемые в автоматизированных системах;
- принципы построения и функционирования, архитектуру, примеры реализаций современных систем управления базами данных;
- методы, способы и средства обеспечения отказоустойчивости автоматизированных систем

уметь:

- анализировать программные, архитектурно-технические и схемотехнические решения компонентов автоматизированных систем
- выявлять потенциальные уязвимости информационной безопасности автоматизированных систем
- разрабатывать предложения по совершенствованию системы управления информационной безопасностью автоматизированных систем;

Тема 1. ИСТОРИЯ РАЗВИТИЯ, НАЗНАЧЕНИЕ И РОЛЬ БАЗ ДАННЫХ. ЭТАПЫ РАЗВИТИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

Первый этап развития СУБД связан с организацией баз данных на больших машинах типа IBM 360/370, ЕС-ЭВМ и мини-ЭВМ типа PDP11 (фирмы Digital Equipment Corporation — DEC), разных моделях HP (фирмы Hewlett Packard).

Базы данных хранились во внешней памяти центральной ЭВМ, пользователями этих баз данных были задачи, запускаемые в основном в пакетном режиме. Интерактивный режим доступа обеспечивался с помощью консольных терминалов, которые не обладали собственными вычислительными ресурсами (процессором, внешней памятью) и служили только устройствами ввода-вывода для центральной ЭВМ. Программы доступа к БД писались на различных языках и запускались как обычные числовые программы.

Особенности этого этапа развития выражаются в следующем:

- Все СУБД базируются на мощных мультипрограммных операционных системах (MVS, SVM, RTE, OSRV, RSX, UNIX), поэтому в основном поддерживается работа с централизованной базой данных в режиме распределенного доступа.
- Функции управления распределением ресурсов в основном осуществляются операционной системой (ОС).
- Поддерживаются языки низкого уровня манипулирования данными, ориентированные на навигационные методы доступа к данным.
- Значительная роль отводится администрированию данных.
- Проводятся серьезные работы по обоснованию и формализации реляционной модели данных, и создается первая система (System R), реализующая идеологию реляционной модели данных.
- Появляются первые языки высокого уровня для работы с реляционной моделью данных. Однако отсутствуют стандарты для этих первых языков.

Особенности второго этапа состоят в следующем:

- Все СУБД были рассчитаны на создание БД в основном с монопольным доступом. И это понятно: компьютер персональный, он не был подсоединен к сети, и база данных на нем создавалась для работы одного пользователя. В редких случаях предполагалась последовательная работа нескольких пользователей, например сначала оператора, который вводил

бухгалтерские документы, а потом главбуха, который определял проводки, соответствующие первичным документам.

- Большинство СУБД имели развитый и удобный пользовательский интерфейс. В основном существовал интерактивный режим работы с БД как в рамках описания БД, так и в рамках проектирования запросов. Кроме того, большинство СУБД предлагали развитый и удобный инструментарий для разработки готовых приложений без программирования. Инструментальная среда состояла из готовых элементов приложения в виде шаблонов экранных форм, отчетов, этикеток (Labels), графических конструкторов запросов, которые достаточно просто могли быть собраны в единый комплекс.
- Во всех настольных СУБД поддерживался только внешний уровень представления реляционной модели, т. е. только внешний табличный вид структур данных.
- В настольных СУБД отсутствовали средства поддержки ссылочной и структурной целостности базы данных. Эти функции должны были выполнять приложения, однако скудость средств разработки приложений иногда не позволяла это сделать, и эти функции должны были выполняться пользователем, требуя от него дополнительного контроля при вводе и изменении информации, хранящейся в БД.
- Наличие монопольного режима работы фактически привело к вырождению функций администрирования БД и в связи с этим — к отсутствию инструментальных средств администрирования БД.

Хорошо известно, что история развивается по спирали, поэтому после процесса «персонализации» начался обратный процесс — интеграция. Множится количество локальных сетей, все больше информации передается между компьютерами, остро встает задача согласованности данных, хранящихся и обрабатывающихся в разных местах, но логически друг с другом связанных, возникают задачи, связанные с параллельной обработкой транзакций — последовательностей операций над БД, переводящих ее из одного непротиворечивого состояния в другое непротиворечивое состояние. Успешное решение этих задач приводит к появлению распределенных баз данных, сохраняющих все преимущества настольных СУБД и в то же время позволяющих организовать параллельную обработку информации и поддержку целостности БД.

Особенности данного этапа состоят в следующем.

- Практически все современные СУБД обеспечивают поддержку полной реляционной модели, а именно:
- структурной целостности — допустимыми являются только данные, представленные в виде отношений реляционной модели;
- языковой целостности, т. е. языков манипулирования данными высокого уровня (в основном SQL);
- ссылочной целостности, контроля за соблюдением ссылочной целостности в течение всего времени функционирования системы, и гарантий невозможности со стороны СУБД нарушить эти ограничения.
- Большинство современных СУБД рассчитаны на многоплатформенную архитектуру, т. е. они могут работать на компьютерах с разной архитектурой и под разными операционными системами, при этом для пользователей доступ к данным, управляемым СУБД на разных платформах, практически неразличим.

Вопросы для подготовки

1. Информация и данные.
2. Базы данных и файловые системы.
3. Функции и состав СУБД.
4. Ранние подходы к организации БД.
5. Понятие базы данных.
6. Файловые системы и системы с базами данных.
7. Компоненты СУБД.
8. Распределение обязанностей в системах с базами данных.
9. Администраторы данных и баз данных, разработчики баз данных, прикладные программисты, пользователи.
10. Классификация задач, решаемых с использованием СУБД.

Тема 2. ОБЩИЕ ПРИНЦИПЫ ПОСТРОЕНИЯ БД. МОДЕЛИ ДАННЫХ

База данных (БД) - именованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области. Под предметной областью понимается некоторая область человеческой деятельности или область реального мира, на основе которой создается БД и её структура.

Система управления базами данных (СУБД) - совокупность языковых и программных средств, предназначенных для создания, наполнения, обновления и удаления баз данных.

Принципы построения баз данных

К современным базам данных, а, следовательно, и к СУБД, на которых они строятся, предъявляются следующие основные требования:

- Высокое быстродействие (малое время отклика на запрос). Время отклика - промежуток времени от момента запроса к БД до фактического получения данных.
- Простота обновления данных.
- Независимость данных - возможность изменения логической и физической структуры БД без изменения представлений пользователей.
- Совместное использование данных многими пользователями.
- Безопасность данных - защита данных от преднамеренного или непреднамеренного нарушения секретности, искажения или разрушения.
- Стандартизация построения и эксплуатации БД (фактически СУБД).
- Адекватность отображения данных соответствующей предметной области.
- Простой интерфейс пользователя.

Важнейшими являются первые два противоречивых требования: повышение быстродействия требует упрощения структуры БД, что, в свою очередь, затрудняет процедуру обновления данных, увеличивает их избыточность.

Безопасность данных включает их целостность и защиту. Целостность данных - устойчивость хранимых данных к разрушению и уничтожению, связанных с неисправностями технических средств, системными ошибками и ошибочными действиями пользователей. Она предполагает:

- отсутствие неточно введенных данных или двух одинаковых записей об одном и том же факте;

- защиту от ошибок при обновлении БД;
- невозможность удаления (или каскадное удаление) связанных данных разных таблиц;
- отсутствие искажение данных при работе в многопользовательском режиме и в распределенных базах данных;
- сохранность данных при сбоях техники (восстановление данных).

Вопросы для подготовки

1. Общая характеристика, назначение и возможности, классификация, состав и архитектура СУБД.
2. Информационное, лингвистическое, математическое, аппаратное, организационное, правовое обеспечения СУБД.
3. Отображение предметной области.
4. Сущности и связи.
5. Методы абстрагирования данных.
6. Иерархическая, сетевая, реляционная модели данных.
7. Трехуровневая архитектура ANSI-SPARC.
8. Внешний уровень. Концептуальный уровень. Внутренний уровень.

Лабораторная работа №1 "Ограничения целостности базы данных".

Цель: знакомство с ограничениями целостности в БД

На основе документации MS SQL Server разработать различные виды механизмов ограничения целостности: структура базы данных, типы, связи, ключи, уникальность, общие ограничения, значения по умолчанию, триггеры.

Сценарий сдачи: скрипты, демонстрирующие работу.

Тема 3. НОРМАЛИЗАЦИЯ БАЗЫ ДАННЫХ

Нормализация — это процесс организации данных в базе данных. Это включает создание таблиц и установку отношений между этими таблицами в соответствии с правилами, предназначенными для защиты данных и обеспечения большей гибкости базы данных за счет исключения избыточности и несогласованности зависимости.

Избыточные данные отнимают место на диске и создают проблемы с обслуживанием. Если необходимо изменить данные, которые находятся в нескольких местах, их необходимо изменить точно так же, как во всех местах. Изменение адресов клиентов значительно упрощается, если эти данные хранятся только в таблице Customers, а не в базе данных.

Существует несколько правил нормализации баз данных. Каждое правило называется "обычной формой". Если выполняется первое правило, база данных называется "Первая нормальная форма". Если выполняются первые три правила, база данных считается "третьей нормальной форме". Хотя возможны и другие уровни нормализации, Третья нормальная форма считается самым высоким уровнем, необходимым для большинства приложений.

Как и в случае с множеством формальных правил и спецификаций, реальные сценарии не всегда позволяют обеспечить оптимальное соответствие требованиям. Как правило, нормализация требует дополнительных таблиц, и некоторые клиенты находят эту громоздкий. Если вы решите, что вы нарушаете одно из первых трех правил нормализации, убедитесь, что приложение предвидит все возможные проблемы, такие как избыточные данные и несогласованные зависимости.

Первая нормальная форма

- Исключите повторяющиеся группы в отдельных таблицах.
- Создайте отдельную таблицу для каждого набора связанных данных.
- Идентифицируйте каждый набор связанных данных с помощью первичного ключа.

Вторая нормальная форма

- Создайте отдельные таблицы для наборов значений, которые применяются к нескольким записям.
- Свяжите эти таблицы с помощью внешнего ключа.

Третья нормальная форма

- Исключите поля, которые не зависят от ключа.

Другие формы нормализации

Четвертая обычная форма, также называемая обычной формой бойце Кодд (БКНФ) и пятая обычная форма, существует, но редко рассматривается в практическом проекте. Отсутствие учета этих правил может привести к меньшему созданию структуры базы данных, но не влияет на функциональность.

Вопросы для подготовки

1. Аномалии при эксплуатации баз данных.
2. Нормальные формы БД.
3. Денормализация.

Лабораторная работа №2 "Нормализация баз данных. 3НФ"

Цель: знакомство с нормальными формами низкого порядка

Создать базу данных по произвольной предметной области (универсальное отношение). Приведите базу данных к 3НФ.

Сценарий сдачи: скрипты, демонстрирующие работу.

Тема 4. ОСНОВЫ ПОСТРОЕНИЯ РЕЛЯЦИОННЫХ БД

Понятие реляционный (англ, relation — отношение) связано с разработками известного американского специалиста в области систем баз данных Е. Кодда.

Эти модели характеризуются простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных.

Реляционная модель ориентирована на организацию данных в виде двухмерных таблиц. Каждая реляционная таблица представляет собой двухмерный массив и обладает следующими свойствами:

• каждый элемент таблицы — один элемент данных;

• все столбцы в таблице однородные, т. е. все элементы в столбце имеют одинаковый тип (числовой, символьный и т. д.) и длину;

• каждый столбец имеет уникальное имя;

• одинаковые строки в таблице отсутствуют;

• порядок следования строк и столбцов может быть произвольным.

Отношения представлены в виде таблиц, строки которых соответствуют кортежам или записям, а столбцы — атрибутам отношений, доменам, полям.

При проектировании базы данных стремятся к достижению эффективности, что предполагает минимальное дублирование данных, удобство их обработки и обновления. Для удовлетворения этих требований необходимо определить состав отношений, включаемых в БД, и какие атрибуты должны входить в эти отношения.

Вопросы для подготовки

1. История реляционной модели.
2. Реляционная модель. Терминология.
3. Структура реляционных данных.
4. Реляционные ключи. Реляционная целостность. Реляционная алгебра. Реляционное исчисление.
5. Представления.

Тема 5. ФИЗИЧЕСКАЯ ОРГАНИЗАЦИЯ БАЗ ДАННЫХ

Физическая модель базы данных определяет способ размещения данных на носителях (устройствах внешней памяти), а также способ и средства организации эффективного доступа к ним. Поскольку СУБД функционирует в составе и под управлением операционной системы и база данных размещается обычно на устройствах общего доступа (разделяемый ресурс), используемого самой ОС и другими прикладными программами, то организация хранения данных и доступа к ним в значительной степени зависит от принципов и методов управления данными операционной системы. И, естественно, СУБД в той или иной степени использует не только файловую систему и подсистему ввода-вывода ОС, но и специализированные методы доступа, основанные на тех или иных принципах организации данных.

С общепринятой точки зрения к вопросам организации данных относятся

- выбор типа записи – единицы обмена в операциях ввода-вывода;
- выбор способа размещения записей в файле и, возможно, метода оптимизации размещения;
- выбор способа адресации и метода доступа к записям.

Целесообразность выделения именно таких аспектов организации была предельно очевидна на начальной стадии развития таких запись-ориентированных систем и устройств внешней памяти, как магнитные ленты и диски. Но, следует отметить, что широкое использование современных поток-ориентированных систем ввода-вывода не уменьшило принципиальное, да и практическое, значение давно известных методов и решений, построенных на запись-ориентированных принципах. Основные понятия и подходы к физической организации и обработке данных иллюстрируются на слайде

Логическая запись, с которой работает прикладная программа – это совокупность элементов или агрегатов данных, воспринимаемая и, обычно, физически отдельно размещаемая в рабочей области памяти прикладной программой как единое целое. Последовательность записей в логике обработки образует файл.

Физическая запись, с которой работает файловая система — это совокупность данных, которые размещаются в файле обычно на внешнем носителе, и могут быть считаны или записаны как единое целое одной командой ввода-вывода.

Здесь **файл** – это последовательность физических записей, размещаемых в линейном пространстве носителя, но, в общем случае, не обязательно в линейном порядке.

Организация данных в случаях логического и физического представления может не совпадать, в частности, одна физическая запись может включать несколько логических (блокирование записей). При этом алгоритмы выделения логических записей из физической в значительной степени зависят от типа записи, рассматриваемого как характер организации последовательности байтов.

Вопросы для подготовки

1. Структуры данных и базы данных.
2. Способы хранения информации в базах данных.
3. Способы повышения эффективности обработки данных за счет их организации.

Лабораторная работа №3 "Нормализация баз данных. 5НФ"

Цель: знакомство с нормальными формами более высокого прядка

Привести созданную в предыдущей лабораторной работе базу данных к 5 НФ.

Сценарий сдачи: скрипты, демонстрирующие работу.

Тема 6. ЯЗЫКОВЫЕ СРЕДСТВА СУБД ДЛЯ РАЗЛИЧНЫХ МОДЕЛЕЙ ДАННЫХ. ЯЗЫК SQL

Языковые средства СУБД представляют собой:

1. Язык описания данных (Data Definition Language, DDL)
2. Язык манипулирования данными (Data Manipulation Language, DML)

DDL позволяет:

- Определить структуру данных
- Определить связи между данными
- Определить ограничения на данные.

DME позволяет описать алгоритмы доступа и обработки данных.

Реализации языковых средств СУБД делятся на:

1. Закрытые системы: со свои собственным языком и не работающих со стандартными языками (dBase, FoxPro)
2. Открытые системы: нет своего языка, а пользующие универсальные языки программирования.
3. Комбинированные системы: симбиоз открытых систем и закрытых. Базовый язык описывает структуры и алгоритмы обработки данных Расширение языка работает с данными.

Способы реализации базового языка:

1. Новый транслятор (Pascal -> Delphi)
2. Препроцессор (программа, реализующая команды работы с данными в последовательности команд базового языка)
3. Call-интерфейс (библиотека, нужная для работы с данными)
   Реализация call-интерфейса через много подпрограмм (dBase: dbopen (), dbclose()…)
   Реализация call-интерфейса через одну процедуру, где команда передается через параметры

SQL (Structured Query Language)

Реляционно-ориентированный язык, позволяющий минимумом команд реализовать около 30 операций по работе с данными, позволяющий как формулировать запросы, так и писать прикладные программы.

Команды делятся на **группы**:

1. Команды определения данных
    1. создание БД
    2. создание таблиц
    3. задание полей таблиц
    4. создание индексов
    5. удаление таблиц, индексов, БД.
2. Запросы на выборку данных
3. Команды модификации данных
    1. добавление данных
    2. удаление данных
    3. изменение данных
4. Команды управления данными
    1. привилегии
    2. параллельный доступ
    3. транзакции

Кроме команд, существует множество **операторов**:

- Арифметические вычисления
- сравнения
- создание виртуальных таблиц
- запоминание результатов запросов/вычислений в таблицах БД
- группировка данных по группам

Типы данных в SQL:

Существует 3 **уровня языка** SQL

1. Минимальный SQL
    - INSERT
    - SELECT

- UPDATE

2. Базовый SQL
    - ALTER TABLE
    - CREATE TABLE
    - CREATE VIEW
    - Более расширенный синтаксис SELECT
3. Расширенный SQL (используется в “навороченных” СУБД Oracle, DB2…)

Вопросы для подготовки

1. Языковые средства манипулирования данными в реляционных СУБД.

2. Язык SQL.

3. Оператор SELECT.

4. Запросы с использованием данных одной таблицы.

5. Реализация операций селекции и проекции.

6. Вычисляемые поля, именование столбцов.

7. Сортировка и группировка результатов запроса.

8. Агрегатные функции.

9. Операции соединения: внутреннее соединение, внешние соединения.

10. Подзапросы. Использование предикатов в подзапросах.

Тема 7. ПЛАНИРОВАНИЕ, ПРОЕКТИРОВАНИЕ И АДМИНИСТРИРОВАНИЕ БД

Жизненный цикл базы данных – это процесс проектирования, реализации и поддержки базы данных. ЖЦБД состоит из семи этапов:

1. предварительное планирование;
2. проверка осуществимости;
3. определение требований;
4. концептуальное проектирование;
5. логическое проектирование;
6. физическое проектирование;
7. оценка работы и поддержка базы данных

Предварительное планирование базы данных – важный этап в процессе перехода от разрозненных данных в интегрированным. На этом этапе собирается информация об используемых и находящихся в процессе разработки прикладки программах и файлах, связанных с ними. Она помогает установить свази между текущими приложениями и то, как используется их информация. Кроме того, позволяет определить будущие требования к базе данных. Информация документируется в виде обобщенной концептуальной модели данных.

Проверка осуществимости предполагает подготовку отчетов по трем вопросам:

- есть ли технология – необходимое оборудование и программное обеспечение – для реализации запланированной базы данных (технологическая осуществимость);
- имеются ли персонал, средства и эксперты для успешного осуществления плана создания базы данных (операционная осуществимость);

- окупится ли запланированная база данных (экономическая эффективность)

Определение требований. На этом этапе определяются:

- цели базы данных;
- информационные потребности различных структурных подразделений и их руководителей;
- требования к оборудованию;
- требования к программному обеспечению.

Концептуальное проектирование. На этом этапе создаются подробные модели пользовательских представлений данных предметной области. Затем они интегрируются в концептуальную модель, которая фиксирует все элементы корпоративных данных, подлежащих загрузке в базу данных. Эту модель ещё называются концептуальной схемой базы данных.

Логическое проектирование. На этом этапе осуществляется выбор типа модели данных. Концептуальная модель отображается в логическую модель, основанную уже на структурах, характерных для выбранной модели.

Физическое проектирование. На этом этапе логическая модель расширяется характеристиками, необходимыми для определения способов физического хранения базы данных, типа устройств для хранения, методов доступа к данных базы, требуемого объема памяти, правил сопровождения базы данных и др.

Оценка и поддержка базы данных. Оценка включает опрос пользователей на предмет выяснения, какие их информационные потребности остались неучтенными. При необходимости в спроектированную базу данных. По мере расширения и изменения потребностей бизнеса поддержка базы данных обеспечивается путем внесения изменений, добавления новых

данных, разработки новых прикладных программ, работающих с базой данных.

Вопросы для подготовки

8. Жизненный цикл приложения баз данных.

9. Этапы жизненного цикла приложения БД.

10.Обзор процедуры проектирования БД.

11.Проектирование приложений.

12.Выбор СУБД.

13.Особенности средств управления в реализациях реляционных СУБД.

14.Администрирование.

Лабораторная работа №4 "Временные таблицы. Функции. процедуры"

Цель: знакомство с процедурным расширением T-SQL

Изучить работу с глобальными и локальными временными таблицами.
Создать процедуры и функции различных типов.

Сценарий сдачи: скрипты, демонстрирующие работу.

Тема 8. ТЕХНОЛОГИЯ И МОДЕЛИ АРХИТЕКТУРЫ КЛИЕНТ/СЕРВЕР

Клиент (Client) – программа, обеспечивающая пользователю доступ к ресурсам на удаленном компьютере, являющемся сервером.

"Толстый" клиент – это наиболее часто встречающийся вариант реализации архитектуры "клиент-сервер" в уже внедренных и активно используемых системах. Такая модель подразумевает объединение в клиентском приложении как презентационной логики, так и бизнес-логики. Серверная часть при описанном подходе представляет собой сервер БД, реализующий логику доступа к ресурсам. К описанной модели часто применяют аббревиатуру RDA – Remote Data Access.

"Тонкий" клиент (thin client) – это компьютер-клиент сети с архитектурой "клиент-сервер", который переносит большинство задач по обработке информации на сервер. Эта модель активно используется в корпоративной среде в связи с распространением интернет-технологий и в первую очередь веб-браузеров. В этом случае клиентское приложение обеспечивает реализацию презентационной логики, а сервер объединяет бизнес-логику и логику доступа к ресурсам. "Тонкие" клиенты лучше использовать для работы с традиционными офисными приложениями.

Двухзвенная модель (two-tier model) – это система "клиент-сервер", в которую входят компьютеры клиента и сервера. Клиент запрашивает данные у сервера, а сервер предоставляет данные. Большинство систем "клиент-сервер" построены с использованием этой модели, но двухзвенные модели способны обеспечить работу лишь ограниченного числа клиентов.

Двухзвенная модель "клиент-сервер" подходит для небольших программ на уровне рабочей группы при числе пользователей менее 100 (конечно, в зависимости от того, что делают прикладные программы). В большинстве двухзвенных систем невозможно существенно увеличить это число.

Многозвенная модель (three-tier model) – это система "клиент-сервер", в которой промежуточное звено (компьютер) помещается между компьютером-клиентом и компьютером-сервером двухзвенной модели. Промежуточное звено, обычно работающее как монитор обработки транзакций или брокер объектных запросов, предоставляет другое место для выполнения программы. С помощью многозвенной модели разработчики могут обеспечивать работу большего числа клиентов, чем при использовании двухзвенной модели.

В компьютерных сетях для передачи данных между узлами сети можно использовать три технологии: коммутацию каналов, коммутацию сообщений и коммутацию пакетов.

Коммутация каналов, обеспечиваемая телефонной сетью общего пользования, позволяет с помощью коммутаторов установить прямое соединение между узлами сети.

При **коммутации сообщений** устройства, называемые коммутаторами и выполненные на базе универсальных или специализированных компьютеров, позволяют накапливать (буферизировать) сообщения и посылать их в соответствии с заданной системой приоритетности и принципами маршрутизации другим узлам сети. Использование коммутации сообщений может увеличить время доставки сообщений по сравнению с коммутацией каналов, однако при этом сглаживаются пиковые нагрузки в сети и повышается живучесть сети.

При **коммутации пакетов** данные пользователя разбиваются на более мелкие порции – пакеты, причем каждый пакет содержит служебные поля и поле данных. Существуют два основных способа передачи данных при пакетной коммутации: виртуальный канал, когда между узлами устанавливается и поддерживается соединение как бы по выделенному каналу (хотя на самом деле физический канал передачи данных разделен между несколькими пользователями) и дейтаграммный режим, когда каждый пакет из набора пакетов, содержащего данные пользователя, передается между

узлами независимо друг от друга. Первый способ соединения называют также контактным режимом (connection mode), второй – бесконтактным (connectionless mode).

Вопросы для подготовки

1. Серверы баз данных.
2. Достоинства и недостатки моделей архитектуры клиент/сервер и их влияние на функционирование сетевых СУБД.
3. Использование средств прямого ввода-вывода, управления памятью, поддержания целостности, защиты от сбоев.
4. Технология оперативной обработки транзакции (OLTP–технология).
5. Поддержка Internet технологий.
6. Оценка эффективности и адаптации функционирования сервера баз данных.
7. Проблемы оптимизации доступа к базе данных.
8. Перспективы развития СУБД.

Тема 9. БЕЗОПАСНОСТЬ БД, УГРОЗЫ, ЗАЩИТА

Современные проблемы обеспечения безопасности БД

Список основных уязвимостей хранилищ данных, актуальный на сегодняшний день, не претерпел существенных изменений за последние более чем пять лет. Проанализировав средства обеспечения безопасности СУБД, архитектуру БД, интерфейсы, известные уязвимости и инциденты безопасности, можно выделить следующие причины возникновения такой ситуации:

- проблемами безопасности серьезно занимаются только крупные производители прежде всего в ведущих продуктах линеек для хранения данных;
- программисты БД, прикладные программисты и администраторы БД не уделяют должного внимания вопросам безопасности;
- разные масштабы и виды хранимых данных требуют разных подходов к безопасности;
- различные СУБД используют разные языковые диалекты для доступа к данным, организованным на основе одной и той же модели;
- появляются новые виды и модели хранения данных.

Особенности систем БД как объекта защиты

В связи с появлением новых решений в области не реляционных хранилищ, размывающих границу традиционного представления о СУБД (например, система кэширования данных в памяти MemcasheDB или Hadoop HDFS), определим функции, отличающие СУБД от файлового хранилища и других типов программных продуктов. В этом ключе в [20] выделено несколько признаков. Переформулировав первый признак – «поддержание логически согласованного набора файлов», в силу активного развития in memory СУБД, осуществляющих хранение и все операции над данными в оперативной памяти, приведем эти критерии в следующей редакции:

- поддержание логически согласованного набора данных;
- обеспечение языка манипулирования данными;
- восстановление информации после разного рода сбоев;
- реальная параллельная работа нескольких пользователей (процессов)

Требования к безопасности БД

Таким образом, на основании разделения уязвимостей можно выделить зависимые и независимые от данных меры обеспечения безопасности хранилищ информации. Независимыми от данных можно назвать следующие требования к безопасной системе БД.

Функционирование в доверенной среде

Под доверенной понимается информационная среда, интегрирующая совокупность защитных механизмов, которые обеспечивают обработку информации без нарушения политики безопасности [21]. В данном случае СУБД должна функционировать в доверенной информационной системе с соответствующими методами обмена данными.

Организация физической безопасности файлов данных

Данный вопрос требует более детального изучения, так как применяемые структуры данных в различных моделях данных СУБД могут иметь значение при шифровании и защите файлов данных. Однако в первом приближении вопрос физической безопасности файлов данных сходен с вопросом физической безопасности любых других файлов пользователей и приложений.

Организация безопасной и актуальной настройки СУБД

К данному аспекту относятся такие общие вопросы обеспечения безопасности, как своевременная установка обновлений, отключение неиспользуемых модулей или применение эффективной политики паролей.

Следующие требования можно назвать зависимыми от данных.

Безопасность пользовательского слоя ПО

К этой категории относятся задачи построения безопасных интерфейсов и вызовов (в том числе с учетом интерфейса СУБД и механизма доступа к данным).

Безопасная организация данных и манипулирование ими

Вопрос организации данных и управления ими является ключевым в системах хранения информации. Несмотря на то, что в приведенном перечне он указан последним, именно в эту область входят задачи организации данных с контролем целостности, обеспечение защиты от логического вывода и другие, специфичные для СУБД проблемы безопасности. Фактически эта задача включает в себя основной пул зависимых от данных уязвимостей и защиты от них.

Вопросы для подготовки

1. Понятие безопасности БД.
2. Угрозы безопасности БД: общие и специфичные.
3. Требования безопасности БД.
4. История развития, назначение и роль баз данных.
5. Модели данных.
6. Математические основы построения реляционных СУБД.

Лабораторная работа №5 "Виды первичных ключей. Индексы. Оптимизация запросов"

Цель: знакомство с механизмами оптимизации запросов

Создать различные виды первичных ключей и индексов. Просмотреть план запросов.

Сценарий сдачи: скрипты, демонстрирующие работу.

Тема 10. КРИТЕРИИ ЗАЩИЩЕННОСТИ БД

Критерии оценки безопасности компьютерных систем (TCSEC), получившие неформальное "Оранжевая книга", были разработаны и опубликованы Министерством обороны США в 1983 г. с целью определения требований безопасности, предъявляемых к аппаратному, программному и специальному программному и информационному обеспечению компьютерных систем, и выработки методологии и технологии анализа степени поддержки политики безопасности в компьютерных системах в основном военного назначения.

Согласно "Оранжевой книге" безопасная компьютерная система — это система, поддерживающая управление доступом к обрабатываемой в ней информации так, что только соответствующим образом авторизованные пользователи или процессы (субъекты), действующие от их имени, получают возможность читать, записывать, создавать и удалять информацию.

Классы защищенности компьютерных систем по TCSEC

"Оранжевая книга" предусматривает четыре группы критериев, которые соответствуют различной степени защищенности: от минимальной (группа D) до формально доказанной (группа A). Каждая группа включает один или несколько классов. Группы D и A содержат по одному классу (классы D и A соответственно), группа C-классы C1, C2, а группа B три класса B1, B2, B3, характеризующиеся различными наборами требований защищенности. Уровень защищенности возрастает от группы D к группе A, а внутри группы - с увеличением номера класса. Усиление требований осуществляется с постепенным смещением акцентов от положений, определяющих наличие в системе каких-то определенных механизмов защиты, к положениям обеспечивающих высокий уровень гарантий того, что система функционирует в соответствии требованиям политики безопасности.

Рассмотрим основные требования классов защищенности по указанным выше четырем категориям:

- политика безопасности;
- подотчетность;
- гарантии;
- документация.

Центральным объектом исследования и оценки по TCSEC является доверительная база вычислений (TCB).

Группа D. Минимальная защита

Класс D. Минимальная защита. Класс D зарезервирован для тех систем, которые были представлены на сертификацию (оценку), но по какой-либо причине ее не прошли.

Группа C. Дискреционная защита

Группа C характеризуется наличием дискреционного управления доступом и аудитом действий субъектов.

Класс C1. Системы на основе дискреционного разграничения доступа. TCB систем, соответствующих этому классу защиты, удовлетворяет неким минимальным требованиям безопасного разделения пользователей и данных. Она определяет некоторые формы разграничения доступа на индивидуальной основе, т.е. пользователь должен иметь возможность защитить свою информацию от ее случайного чтения или уничтожения. Пользователи могут обрабатывать данные как по отдельности, так и от имени группы пользователей

Класс C1 рассчитан на многопользовательские системы, в которых осуществляется совместная обработка данных одного уровня конфиденциальности.

Класс C2. Системы, построенные на основе управляемого дискреционного разграничения доступа.

Системы, сертифицированные по данному классу, должны удовлетворять всем требованиям, изложенным в классе C1. Однако, системы класса C2 поддерживают более тонкую, чем в классе C1, политику дискреционного разграничения доступа, делающую пользователя индивидуально ответственным за свои действия после процедуры аутентификации в системе, а также аудит событий, связанных с безопасностью системы.

Группа B. Мандатное управление доступом.

Основные требования этой группы - мандатное (полномочное) управление доступом с использованием меток безопасности, реализация некоторой формальной модели политики безопасности, а также наличие спецификаций на функции ТСВ. В системах этой группы постепенно к классу B3 должен быть реализован монитор ссылок, который должен контролировать все доступы субъектов к объектам системы.

Класс B1. Системы класса B1 должны удовлетворять требованиям класса C2. Кроме того, должны быть выполнены следующие дополнительные требования. B2

Класс B2. Структурированная защита. Выполняются все требования класса защиты B1. Кроме того, в системах класса B2 ТСВ основывается на четко определенной и хорошо документированной формальной модели политики безопасности, требующей, чтобы мандатная и дискреционная системы разграничения доступа были распространены на все субъекты и объекты компьютерной системы. ТСВ должна быть четко структурирована на элементы, критичные с точки зрения безопасности и некритичные. Интерфейс ТСВ должен быть хорошо определен и ее проект, и конечный результат

должны быть подвергнуты полной проверке и тестированию. Механизм аудита должен быть усилен, введен контроль за конфигурацией системы. Система должна быть устойчива к внешнему проникновению.

Класс В3. Домены безопасности. В системах класса В3 ТСВ должна удовлетворять всем требованиям предыдущего класса и дополнительно требованиям монитора ссылок, который должен быть:

- защищен от несанкционированного изменения или порчи;
- обрабатывать все обращения;
- прост для анализа и тестирования.

ТСВ должна быть структурирована таким образом, чтобы исключить код, не имеющий отношения к безопасности системы. Дополнительно должно быть обеспечено:

- поддержка администратора безопасности;
- расширение механизма аудита с целью сигнализации о любых событиях, связанных с безопасностью;
- поддержка процедуры восстановления системы.

Группа А. Верифицированная защита.

Данная группа характеризуется применением формальных методов верификации корректности работы механизмов управления доступом (дискреционного и мандатного). Требуется, чтобы было формально показано соответствие архитектуры и реализации ТСВ требованиям безопасности.

Класс А1. Формальная верификация. Критерий защиты класса А1 не определяет дополнительные по сравнению с классом В3 требования к архитектуре или политике безопасности компьютерной системы. Дополнительным свойством систем, отнесенных к классу А1, является проведенный анализ ТСВ на соответствие формальным высокоуровневым

спецификациям и использование технологий проверки с целью получения высоких гарантий того, что TCB функционирует корректно.

Вопросы для подготовки

1. Критерии оценки надежных компьютерных систем (TCSEC).
2. Понятие политики безопасности.
3. Совместное применение различных политик безопасности в рамках единой модели.
4. Интерпретация TCSEC для надежных СУБД (TDI).
5. Оценка надежности СУБД как компоненты вычислительной системы.

Тема 11. МОДЕЛИ БЕЗОПАСНОСТИ В СУБД

Модель безопасности включает:

- модель компьютерной (информационной) системы;
- критерии, принципы, ограничения и целевые функции защищенности информации от угроз;
- формализованные правила, ограничения, алгоритмы, схемы и механизмы безопасного функционирования системы.

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

Определяются два основополагающих принципа безопасности функционирования информационных систем:

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

Соответственно в структуре ядра СУБД выделяется дополнительный компонент, называемый **монитором** (сервером, менеджером, ядром) **безопас ности** (Trusted Computing Base-- TCB), который реализует определенную **политику безопасности** во всех процессах обработки данных. Если в схемотехническом аспекте компьютерную систему представить как совокупность ядра, включающего компоненты представления данных и доступа (манипулирования) к данным, а также надстройки, которая реализует интерфейсные и прикладные функции.

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

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).

Вопросы для подготовки

1. Дискреционная (избирательная) и мандатная (полномочная) модели безопасности.
2. Классификация моделей.
3. Аспекты исследования моделей безопасности.
4. Особенности применения моделей безопасности в СУБД.

Тема 12. СРЕДСТВА ИДЕНТИФИКАЦИИ И АУТЕНТИФИКАЦИИ

Идентификацию и аутентификацию можно считать основой программно-технических средств безопасности, поскольку остальные сервисы рассчитаны на обслуживание именованных субъектов. Идентификация и аутентификация — это первая линия обороны, "проходная" информационного пространства организации.

Идентификация позволяет субъекту (пользователю, процессу, действующему от имени определенного пользователя, или иному аппаратно-программному компоненту) назвать себя (сообщить свое имя). Посредством аутентификации вторая сторона убеждается, что субъект действительно тот, за кого он себя выдает. В качестве синонима слова "аутентификация" иногда используют словосочетание "проверка подлинности".

Аутентификация бывает **односторонней** (обычно клиент доказывает свою подлинность серверу) и **двусторонней** (взаимной).

Пример односторонней аутентификации - процедура входа пользователя в систему.

В сетевой среде, когда стороны идентификации/аутентификации террит ориально разнесены, у рассматриваемого сервиса есть два основных аспекта:

- что служит аутентификатором (то есть используется для подтверждения подлинности субъекта);
- как организован (и защищен) обмен данными идентификации/аутентификации.

Субъект может подтвердить свою подлинность, предъявив по крайней мере одну из следующих сущностей:

- нечто, что он знает (пароль, личный идентификационный номер, криптографический ключ и т.п.);
- нечто, чем он владеет (личную карточку или иное устройство аналогичного назначения);

- нечто, что есть часть его самого (голос, отпечатки пальцев и т.п., то есть свои биометрические характеристики).

В открытой сетевой среде между сторонами идентификации/аутентификации не существует доверенного маршрута; это значит, что в общем случае данные, переданные субъектом, могут не совпадать с данными, полученными и использованными для проверки подлинности. Необходимо обеспечить защиту от пассивного и активного прослушивания сети, то есть от перехвата, изменения и/или воспро изведения данных. Передача паролей в открытом виде, очевидно, неудовлетворительна; не спасает положение и шифрование паролей, так как оно не защищает от воспроизведения. Нужны более сложные протоколы аутентификации.

Надежная идентификация и затруднена не только из-за сетевых угроз, но и по целому ряду причин. Во- первых, почти все аутентификационные су щности можно узнать, украсть или подделать. Во- вторых, имеется противор ечие между надежностью аутентификации, с одной стороны, и удобствами пользователя и системного администратора с другой. Так, из соображений безопасности необходимо с определенной частотой просить пользователя повторно вводить аутентификационную информацию (ведь на его место мог сесть другой человек), а это не только хлопотно, но и повышает вероятность того, что кто-то может подсмотреть за вводом данных. В-третьих, чем надежнее средство защиты, тем оно дороже.

Современные средства идентификации/аутентификации должны поддерживать концепцию единого входа в сеть. Единый вход в сеть — это, в первую очередь, требование удобства для пользователей. Если в корпоративной сети много информационных сервисов, допускающих независимое обращение, то многократная идентификация/аутентификация ст ановится слишком обременительной. К сожалению, пока нельзя сказать, что единый вход в сеть стал нормой, доминирующие решения пока не сформировались.

Таким образом, необходимо искать компромисс между надежностью, доступностью по цене и удобством использования и администрирования средств идентификации и аутентификации.

Любопытно отметить, что сервис идентификации / аутентификации может стать объектом атак на доступность. Если система сконфигурирована так, что после определенного числа неудачных попыток устройство ввода идентификационной информации (такое, например, как терминал) блокируется, то злоумышленник может остановить работу легального пользователя буквально несколькими нажатиями клавиш.

Вопросы для подготовки

1. Общие сведения.
2. Совместное применение средств идентификации и аутентификации, встроенных в СУБД и в ОС.

Лабораторная работа №6 "Аутентификация на уровне ОС и на уровне СУБД"

Создать имена вход и пользователей базы данных. Изучить олицетворение.

1. Для доступа к SQL Server создайте 4 учетные записи (логины): «Администратор БД», «Сотрудник отдела кадров», «Сотрудник отдела продаж», «Сотрудник отдела поставок»;

2. Учетную запись «Администратор» наделите привилегиями системного администратора (с помощью системной роли);

3. Напишите SQL-скрипты для получения следующей информации:

3.1. Секретный идентификатор, имя, хэш пароля определенной учетной записи;

3.2. Список всех учетных записей сервера;

3.3. Список всех учетных записей сервера, обладающих правами администратора;

4. Напишите SQL-скрипты для выполнения следующих действий с учетной записью SQL-сервера:

4.1. Блокировка учетной записи (временное приостановление действия);

4.2. Разблокировка учетной записи;

4.3. Изменение пароля учетной записи;

4.4. Изменение БД по умолчанию;

4.5 Удаление учетной записи;

5. Напишите SQL-скрипты для выполнения следующих действий с учетной записью операционной системы (ОС):

5.1. Регистрация учетной записи ОС в качестве учетной записи в MS SQL Server;

5.2. Отмена регистрации учетной записи ОС в качестве учетной записи в MS SQL Server;

5.3. Запрет подключений учетной записи ОС в качестве учетной записи в MS SQL Server;

6. Для каждой учетной записи, созданной в 1 пункте, кроме «Администратор БД» добавьте пользователя в вашу БД (AdventureWorks2008R2);

Сценарий сдачи: скрипты, демонстрирующие работу.

Тема 13. СРЕДСТВА УПРАВЛЕНИЯ ДОСТУПОМ

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

- добровольное управление доступом;
- принудительное управление доступом.

При добровольном управлении доступом вводится так называемое **владение объектами.** Как правило, владельцами объектов являются те субъекты базы данных, процессы которых создали соответствующие объекты. Добровольное управление доступом заключается в том, что права на доступ к объектам определяют их владельцы. Иначе говоря, соответствующие ячейки матрицы доступа заполняются теми субъектами (пользователями), которым принадлежат права владения над соответствующими объектами базы данных. В большинстве систем права владения объектами могут передаваться. В результате при добровольном управлении доступом реализуется полностью децентрализованный принцип организации и управления процессом разграничения доступа.

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

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

Принудительный способ обеспечивает более жесткое централизованное управление доступом. Вместе с тем он является менее гибким и менее точным в плане настройки системы разграничения доступа на потребности и

полномочия пользователей, так как наиболее полное представление о содержимом и конфиденциальности объектов (ресурсов) имеют, соответственно, их владельцы.

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

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

Вопросы для подготовки

1. Основные понятия: субъекты и объекты, группы пользователей, привилегии, роли и представления.
2. Виды привилегий: привилегии безопасности и доступа.
3. Использование ролей и привилегий пользователей.
4. Соотношение прав доступа, определяемых ОС и СУБД.

5. Использование представлений для обеспечения конфиденциальности информации в СУБД.
6. Средства реализации мандатной политики безопасности в СУБД.

Лабораторная работа №7 "Управление доступом к базе данных"

Настроить разрешения для пользователей и имен входов.

1. Используя роли и привилегии каждому пользователю назначьте следующие права:

1.1 «Сотрудник отдела продаж»

a. разрешение на добавление, изменение, удаление данных о сотрудниках компании Adventure Works Cycles;
b. запрет на просмотр, изменение и удаление данных продукции, продаж и поставок;

1.2. «Сотрудник отдела продаж»

a. разрешение на добавление, изменение, удаление данных о заказчиках и продажах;
b. разрешение на просмотр данных о продукции, поставщиках и поставках;
c. запрет на изменение и удаление данных о продукции, поставщиках и поставках;

1.3. «Сотрудник отдела поставок»

a. разрешение на добавление, изменение, удаление данных о поставщиках и поставках;
b. разрешение на изменение информации о количестве продукции на складе компании;
c. для данных о продукции - разрешение на просмотр и запрет на изменение и удаление;
d. запрет на просмотр, изменение и удаление данных о сотрудниках и продажах;

2. Напишите SQL-скрипты для получения следующей информации:

2.1. Все пользователи и все привилегии текущей БД;

2.2. Список всех ролей и пользователей, которым присвоены эти роли;

2.3. Список всех ролей, назначенных текущему пользователю;

2.4. Список привилегий, ассоциированных с какой-то конкретной ролью;

2.5. Список всех привилегий текущего пользователя;

Напишите код запроса с использованием конструкции EXECUTE AS, в ходе которого «Сотрудник отдела поставок» смог бы выполнять запросы от имени пользователя «Сотрудник отдела продаж».

Сценарий сдачи: скрипты, демонстрирующие работу.

Тема 14. ЦЕЛОСТНОСТЬ БД И СПОСОБЫ ЕЕ ОБЕСПЕЧЕНИЯ

Целостность актуальность и непротиворечивость информации, ее защищенность от разрушения и несанкционированного изменения. Целостность данных - неотъемлемое свойство базы данных, и ее обеспечение является важнейшей задачей проектирования БнД. Целостность данных описывается набором специальных предложений, называемых ограничениями целостности. Ограничения целостности представляют собой утверждения о допустимых значениях отдельных информационных единиц и связях между ними. Эти ограничения определяются в большинстве случаев особенностями предметной области, хотя могут отражать и чисто информационные (лингвистические) характеристики.

К основным информационным объектам дополнительно могут накладываться следующие ограничения: Ограничения алгоритмических зависимостей между показателями. Запрет на обновление. Ограничение целостности по моменту контроля. Ограничение целостности по режиму проверки корректности БД Ограничение целостности по необходимости описания. Ограничение целостности служебной информации. Информационная целостность банка данных.

По моменту контроля за соблюдением ограничения целостности различают одномоментные и отложенные ограничения целостности. Отложенные ограничения целостности могут не соблюдаться в процессе выполнения какой-то группы операций, но должны быть соблюдены по их завершении. С понятием отложенного ограничения целостности тесно связано понятие транзакции - законченной совокупности действий над БД, которая переводит БД из одного целостного в логическом смысле состояния в другое целостное состояние. Примером отложенных ограничений целостности могут служить действия при выполнении бухгалтерских проводок: в бухгалтерском учете действует принцип двойной записи; в какой-то момент, когда проведена

запись по дебету счета, но еще не проведена запись по кредиту корреспондирующего счета, может временно нарушиться баланс, но по завершении операции баланс должен соблюдаться.

Другим признаком классификации по временному признаку является классификация по режиму проверки корректности БД. Возможны два режима проверки ограничений целостности: проверка в момент корректировки и проверка существующей БД. Первый из них является оперативным режимом, второй - аудитом БД. По необходимости описания ограничения целостности могут быть явными и неявными. Неявные ограничения целостности определяются спецификой модели данных и проверяются СУБД автоматически.

Понятие целостности может касаться и служебной информации. Для реляционных СУБД это прежде всего относится к поддержанию соответствия между индексными файлами и соответствующими им индексируемыми файлами баз данных. Наряду с понятием целостности базы данных может быть введено понятие информационной целостности банка данных, заключающееся в обеспечении правильности взаимосвязи всех его информационных компонентов.

Некоторые СУБД имеют специальный механизм, позволяющий отслеживать согласованность различных информационных компонентов банка данных. Для отслеживания взаимосвязи между всеми информационными компонентами БнД должны использоваться словари данных. С обеспечением целостности БнД в целом на настоящий момент времени дело обстоит хуже, чем с контролем целостности БД в узком смысле этого понятия.

Вопросы для подготовки

1. Основные виды и причины возникновения угроз целостности.
2. Способы противодействия.
3. Цели использования триггеров.

4. Способы задания, моменты выполнения.
5. Декларативная и процедурная ссылочные целостности.
6. Внешний ключ.
7. Способы поддержания ссылочной целостности.

Тема 15. КЛАССИФИКАЦИЯ УГРОЗ КОНФИДЕНЦИАЛЬНОСТИ СУБД

Классификация по цели реализации угрозы:

- нарушение конфиденциальности информации
- нарушение целостности информации
- полное или частичное нарушение работоспособности

Классификация по природе возникновения угрозы

- естественные угрозы (стихия)
- искусственные угрозы (человек)

Классификация по локализации источника угрозы представляется следующим образом:

Угрозы непосредственным источником которых является человек

- разглашение, передача или утрата атрибутов разграничения доступа (паролей, ключей шифрования)
- подкуп или шантаж
- копирование конфиденциальных данных легальным пользователем (для продажи, шантажа, и тд)
- взлом системы защиты с целью деструктивных действий
- внедрение агентов фирм-конкурентов (инсейдеры)

Угрозы непосредственным источником которых являются штатные программно-аппаратные средства ИС

- неквалифицированное использование или ошибочный ввод параметров программ, способных привести к полной или частичной потере работоспособности системы
- неквалифицированное использование или ошибочный ввод параметров способных привести к необратимым изменениям в системе
- отказы и сбои в работе ОС, СУБД и прикладных программ

Угрозы непосредственным источником которых являются несанкционированно используемые программно-аппаратные средства

- нелегальное внедрение и использование ПО, не являющихся необходимыми для выполнения нарушителем своих служебных обязанностей.
- нелегальное внедрение
- заражение компьютера вирусами
- работа генератора шума

Угрозы непосредственным источником которых является средой обитания

- внезапное и длительное отключение электропитания
- техногенные и природные катастрофы

Угрозы, источник которых расположен в пределах контролируемой зоны, расположения автоматизированной ИС

Угрозы, источник которых имеет доступ к терминальным устройствам автоматизированной ИС

Угрозы, источник которых имеет доступ к помещениям, где расположены серверы автоматизированной информационной системы

Классификация по способу воздействия на методы и средства хранения данных информационной системы

Угрозы нарушения ИБ данных хранимых на внешних ЗУ

- нарушение конфиденциальности
- создание несанкционированных копий файлов

Угрозы нарушения ИБ данных хранимых в ОП серверов и клиентских компьютеров

Угрозы нарушения ИБ данных, отображаемых на терминале пользователя или принтере

Классификация по характеру воздействия на ИС (2 варианта)

- активное воздействие (выполнение действий)

- пассивное воздействие (наблюдение)

Вопросы для подготовки

1. Причины, виды, основные методы нарушения конфиденциальности.
2. Типы утечки конфиденциальной информации из СУБД, частичное разглашение.
3. Получение несанкционированного доступа к конфиденциальной информации путем логических выводов.
4. Методы противодействия.
5. Особенности применения криптографических методов.

Лабораторная работа №9 "Шифрование"

Изучить различные виды шифрования в СУБД.
Сценарий сдачи: скрипты, демонстрирующие работу.

Тема 16. АУДИТ И ПОДОТЧЕТНОСТЬ

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

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

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

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

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

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

Объект **Спецификация аудита сервера** принадлежит аудиту. Для каждого аудита вы можете создать один объект спецификации аудита сервера, поскольку они оба создаются в области экземпляра SQL Server.

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

Объект **Спецификация аудита базы данных** также принадлежит подсистеме аудита SQL Server. Для каждого аудита каждой базы данных SQL Server можно создать одну спецификацию аудита базы данных.

Спецификация аудита базы данных включает действия аудита уровня базы данных, вызываемые компонентом расширенных событий. В спецификацию аудита базы данных можно добавлять либо группы действий аудита, либо события аудита. **События аудита** — это атомарные события, аудит которых может производиться ядром SQL Server. **Группы действий аудита** — это стандартные группы действий. Они расположены в области базы данных SQL Server. Эти действия передаются аудиту, который регистрирует их в целевом объекте. Не включайте объекты области сервера, такие как системные представления, в пользовательскую спецификацию аудита базы данных.

Вопросы для подготовки

1. Подотчетность действий пользователя и аудит связанных с безопасностью событий.
2. Регистрация действий пользователя.
3. Управление набором регистрируемых событий.
4. Анализ регистрационной информации.

Лабораторная работа №9 "Триггеры безопасности. Журналирование в СУБД"

Разработать триггер входа. Изучить механизм аудита.

Сценарий сдачи: скрипты, демонстрирующие работу.

Тема 17. ТРАНЗАКЦИИ И БЛОКИРОВКИ

Под **транзакцией** понимается неделимая с точки зрения воздействия на БД последовательность операторов манипулирования данными (чтения, удаления, вставки, модификации), приводящая к одному из двух возможных результатов: либо последовательность выполняется, если все операторы правильные, либо вся транзакция откатывается, если хотя бы один оператор не может быть успешно выполнен. Обработка транзакций гарантирует целостность информации в базе данных. Таким образом, транзакция переводит базу данных из одного целостного состояния в другое.

Поддержание механизма транзакций – показатель уровня развитости СУБД. Корректное поддержание транзакций одновременно является основой обеспечения целостности БД. Транзакции также составляют основу изолированности в многопользовательских системах, где с одной БД параллельно могут работать несколько пользователей или прикладных программ. Одна из основных задач СУБД – обеспечение изолированности, т.е. создание такого режима функционирования, при котором каждому пользователю казалось бы, что БД доступна только ему. Такую задачу СУБД принято называть параллелизмом транзакций.

Большинство выполняемых действий производится в теле транзакций. По умолчанию каждая команда выполняется как самостоятельная транзакция. При необходимости пользователь может явно указать ее начало и конец, чтобы иметь возможность включить в нее несколько команд.

При выполнении транзакции система управления базами данных должна придерживаться определенных правил обработки набора команд, входящих в транзакцию. В частности, разработано четыре правила, известные

как требования ACID, они гарантируют правильность и надежность работы системы.

ACID-свойства транзакций

Характеристики транзакций описываются в терминах ACID (Atomicity, Consistency, Isolation, Durability – неделимость, согласованность, изолированность, устойчивость).

- Транзакция **неделима** в том смысле, что представляет собой единое целое. Все ее компоненты либо имеют место, либо нет. Не бывает частичной транзакции. Если может быть выполнена лишь часть транзакции, она отклоняется.
- Транзакция является **согласованной**, потому что не нарушает бизнес-логику и отношения между элементами данных. Это свойство очень важно при разработке клиент-серверных систем, поскольку в хранилище данных поступает большое количество транзакций от разных систем и объектов. Если хотя бы одна из них нарушит целостность данных, то все остальные могут выдать неверные результаты.
- Транзакция всегда **изолирована**, поскольку ее результаты самодостаточны. Они не зависят от предыдущих или последующих транзакций
  это свойство называется сериализуемостью и означает, что транзакции в последовательности независимы.
- Транзакция **устойчива.** После своего завершения она сохраняется в системе, которую ничто не может вернуть в исходное (до начала транзакции) состояние, т.е. происходит фиксация транзакции, означающая, что ее действие постоянно даже при сбое системы. При этом подразумевается некая форма хранения информации в постоянной памяти как часть транзакции.

Блокировки

Повышение эффективности работы при использовании небольших транзакций связано с тем, что при выполнении транзакции сервер накладывает на данные блокировки.

Блокировкой называется временное ограничение на выполнение некоторых операций обработки данных. Блокировка может быть наложена как на отдельную строку таблицы, так и на всю базу данных. Управлением блокировками на сервере занимается менеджер блокировок, контролирующий их применение и разрешение конфликтов. Транзакции и блокировки тесно связаны друг с другом. Транзакции накладывают блокировки на данные, чтобы обеспечить выполнение требований ACID. Без использования блокировок несколько транзакций могли бы изменять одни и те же данные.

Блокировка представляет собой метод управления параллельными процессами, при котором объект БД не может быть модифицирован без ведома транзакции, т.е. происходит блокирование доступа к объекту со стороны других транзакций, чем исключается непредсказуемое изменение объекта. Различают два вида блокировки:

- блокировка записи – транзакция блокирует строки в таблицах таким образом, что запрос другой транзакции к этим строкам будет отменен;
- блокировка чтения – транзакция блокирует строки так, что запрос со стороны другой транзакции на блокировку записи этих строк будет отвергнут, а на блокировку чтения – принят.

В СУБД используют протокол доступа к данным, позволяющий избежать проблемы параллелизма. Его суть заключается в следующем:

- транзакция, результатом действия которой на строку данных в таблице является ее извлечение, обязана наложить блокировку чтения на эту строку;
- транзакция, предназначенная для модификации строки данных, накладывает на нее блокировку записи;

- если запрашиваемая блокировка на строку отвергается из-за уже имеющейся блокировки, то транзакция переводится в режим ожидания до тех пор, пока блокировка не будет снята;
- блокировка записи сохраняется вплоть до конца выполнения транзакции.

Вопросы для подготовки

1. Транзакции как средство изолированности пользователей.
2. Сериализация транзакций.
3. Методы сериализации транзакций.
4. Режимы блокировок. Правила согласования блокировок.
5. Двухфазный протокол синхронизационных блокировок.
6. Тупиковые ситуации, их распознавание и разрушение.

Лабораторная работа №10 "Транзакции. Уровни изоляции транзакций"

Цель: знакомство с транзакционным механизмом СУБД

Применить транзакции как средства изолированности пользователей. Режимы блокировок. Правила согласования блокировок. Двухфазный протокол синхронизационных блокировок. Тупиковые ситуации, их распознавание и разрушение.

Сценарий сдачи: скрипты, демонстрирующие работу.

СПИСОК ЛИТЕРАТУРЫ

1. **Мартишин, С. А. Базы данных. Практическое применение СУБД SQL и NoSQL-типа для проектирования информационных систем:** учебное пособие / С.А. Мартишин, В. Л. Симонов, М. В. Храпченко. — Москва: ИД «ФОРУМ»: ИНФРА-М, 2016. — 368 с. — (Высшее образование). - ISBN 978-5-8199-0660-6. - Текст: электронный. - URL: https://znanium.com/catalog/product/556449 (дата обращения: 15.05.2020). – Режим доступа: по подписке
2. **Дадян, Э. Г. Методы, модели, средства хранения и обработки данных:** учебник / Э.Г. Дадян, Ю. А. Зеленков. — Москва: Вузовский учебник: ИНФРА-М, 2017. — 168 с. - ISBN 978-5-9558-0490-3. - Текст: электронный. - URL: https://znanium.com/catalog/product/543943 (дата обращения: 15.05.2020). – Режим доступа: по подписке.
3. **Баранова, Е. К. Информационная безопасность и защита информации:** учеб. пособие / Баранова Е. К., Бабаш А.В. — 3-е изд., перераб. и доп. — Москва: РИОР: ИНФРА-М, 2017. — 322 с. — (Высшее образование). — ww.dx.doi.org/10.12737/11380. - ISBN 978-5-369-01450-9. - Текст: электронный. - URL: https://znanium.com/catalog/product/763644 (дата обращения: 15.05.2020). – Режим доступа: по подписке.

Интернет-ресурсы

1. Вузовские электронно-библиотечные системы учебной литературы.
2. http://www.infosecurity. Report.ru (портал по информационной безопасности).
3. База научно-технической информации ВИНИТИ РАН.
4. Среды разработки на языках C#, C++, Delphi.
5. Порталы разработчиков систем управления базами данных: MS SQL Server, InterBase/FireBird, MySQL, Oracle, SQL Postgre.