МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

Кафедра аппаратно-программных комплексов вычислительной техники Академии ЛИМТУ

Реферат

на тему:

Администрирование баз данных.

Выполнил:

Ильин Андрей Михайлович

Группа: ***5895***

Рецензент:

Юдина Ольга Алексеевна

(Тьютор)

подпись_____

Санкт-Петербург

2015г.

Оглавление.

Введение

Современные базы данных – это сложные многофункциональные программные системы, работающие в открытой распределенной среде. Они уже сегодня доступны для использования в деловой сфере и выступают не просто в качестве технических и научных решений, но как завершенные продукты, предоставляющие разработчикам мощные средства управления данными и богатый инструментарий для создания прикладных программ и систем.

Администрирование базами данных предусматривает выполнение функций, направленных на обеспечение надежного и эффективного функционирования системы баз данных, адекватности содержания базы данных информационным потребностям пользователей, отображения в базе данных актуального состояния предметной области.

Необходимость персонала, обеспечивающего администрирование данными в системе БД в процессе функционирования, является следствием централизованного характера управления данными в таких системах, постоянно требующего поиска компромисса между противоречивыми требованиями к системе в социальной пользовательской среде. Хотя такая необходимость и признавалась на ранних стадиях развития технологии баз данных, четкое понимание и структуризация функций персонала, занятого администрированием, сложилось только вместе с признанием многоуровневой архитектуры СУБД.

Проблеме администрирования баз данных внимание уделяется сравнительно недавно – с появлением и развитием современных баз данных. Однако в связи с тем, что совершенствование баз данных и систем управления данных – явление постоянное и непрерывное, проблема остается достаточно актуальной, следовательно, требует дополнительных исследований в данной области компьютерных технологий.

1.Администратор базы данных – основные понятия

1.1 Понятие, классификация и функции администратора базы данных

Функционирование базы данных (БД) невозможно без участия специалистов, обеспечивающих создание, функционирование и развитие базы данных. Такая группа специалистов называется администратором базы данных (АБД). Эта группа специалистов считается составной частью базы данных.

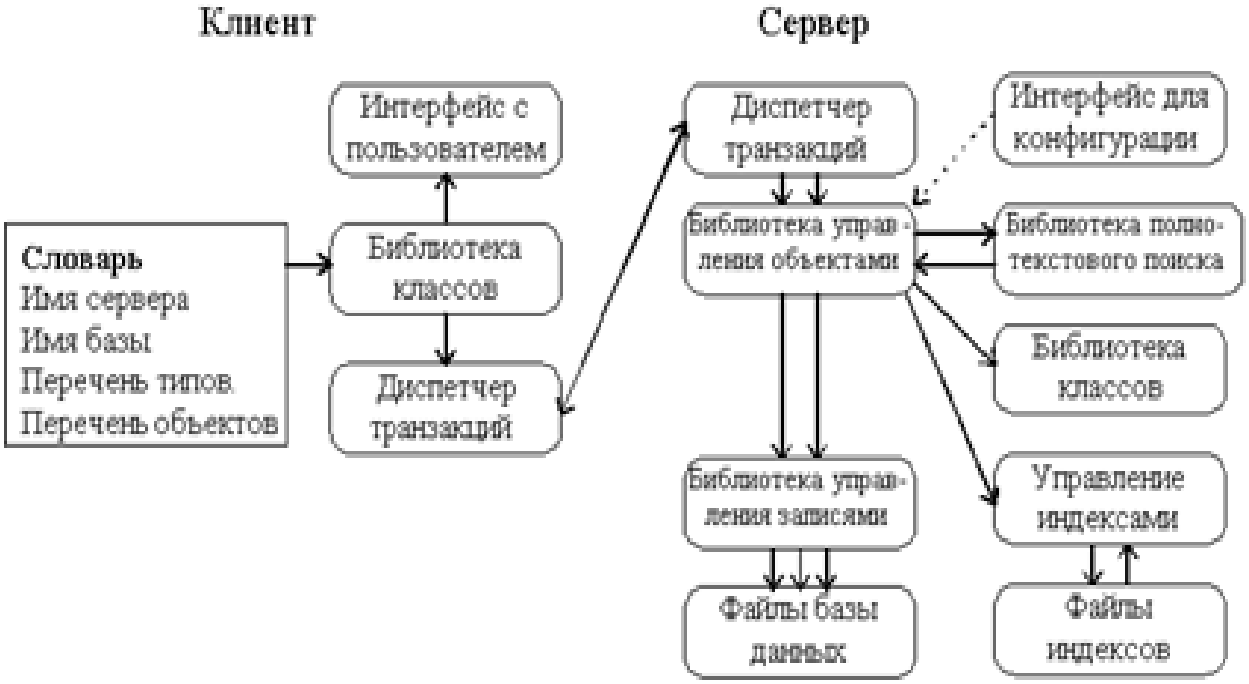

Рис.1 - Обобщенная структура системы управления базой данных

В зависимости от сложности и объема банка данных, от особенностей используемой системы управления базы данных (СУБД), общую схему которой можно увидеть на рис. 1 служба администрации базы данных может различаться как по составу и квалификации специалистов, так и по количеству работающих в этой службе.

Администратор базы данных выполняют работы по созданию и обеспечению функционирования БД на протяжении всех этапов жизненного цикла системы. В составе группы администраторов банка данных можно выделить различные подгруппы в зависимости от выполняемых ими функций. Численность группы администрации, выполняемые ими функции, будут в значительной степени зависеть от масштаба банка данных, специфики хранимой в нем информации, типа банка данных, особенностей используемых программных средств и некоторых других факторов.

В составе администрации базы данных должны быть системные аналитики, проектировщики структур данных и внешнего по отношению к банку данных информационного обеспечения, проектировщики технологических процессов обработки данных, системные и прикладные программисты, операторы, специалисты по техническому обслуживанию. Если речь идет о коммерческом банке данных, то важную роль здесь будут играть специалисты по маркетингу.

Администраторы базы данных выполняют большой круг разнообразных функций:

1. Анализ предметной области: описание предметной области, выявление ограничений целостности, определение статуса информации, определение потребностей пользователей, определение статуса

пользователей, определение соответствия «данные – пользователь», определение объемно-временных характеристик обработки данных.

2. Проектирование структуры базы данных: определение состава и структуры информационных единиц, составляющих базу данных, задание связей между ними, выбор методов упорядочения данных и методов доступа к информации, описание структуры БД на языке обработки данных (ЯОД).

3. Задание ограничений целостности при описании структуры базы данных и процедур обработки БД: задание ограничений целостности, присущих предметной области, определение ограничений целостности, вызванных структурой базы данных, разработка процедур обеспечения целостности БД при вводе и корректировке данных, обеспечение ограничений целостности при параллельной работе пользователей в многопользовательском режиме.

4. Первоначальная загрузка и ведение базы данных: разработка технологии первоначальной загрузки и ведения (изменения, добавления, удаления записей) БД, проектирование форм ввода, создание программных модулей, подготовка исходных данных, ввод и контроль ввода.

5. Защита данных от несанкционированного доступа:

– обеспечение парольного входа в систему: регистрация пользователей, назначение и изменение паролей;

– обеспечение защиты конкретных данных: определение прав доступа групп пользователей и отдельных пользователей, определение допустимых операций над данными для отдельных пользователей, выбор/создание программно-технологических средств защиты данных; шифрование информации с целью защиты данных от несанкционированного использования;

– тестирование средств защиты данных;

– фиксация попыток несанкционированного доступа к информации;

– исследование возникающих случаев нарушения защиты данных и проведение мероприятий по их предотвращению.

6. Защита данных от разрушений. Одним из способов защиты от потери данных является резервирование. Используется как при физической порче файла, так и в случае, если в БД внесены нежелательные необратимые изменения.

7. Обеспечение восстановления БД: разработка программно-технологических средств восстановления БД, организация ведения системных журналов.

8. Анализ обращений пользователей к БД: сбор статистики обращений пользователей к БД, ее хранение и анализ (кто из пользователей, к какой информации, как часто обращался, какие выполнял операции, время выполнения запросов, анализ причин безуспешных (в т.ч. и аварийных) обращений к БД.

9. Анализ эффективности функционирования базы данных и развитие системы: анализ показателей функционирования системы (время обработки,

объем памяти, стоимостные показатели), реорганизация и реструктуризация баз данных, изменение состава баз данных, развитие программных и технических средств.

10. Работа с пользователями: сбор информации об изменениях в предметной области, об оценке пользователями работы базы данных, определение регламента работы пользователей с базой данных, обучение и консультирование пользователей.

11. Подготовка и поддержание системных программных средств: сбор и анализ информации о СУБД и других прикладных программ, приобретение программных средств, их установка, проверка работоспособности, поддержание системных библиотек, развитие программных средств.

12. Организационно-методическая работа: выбор или создание методики проектирования БД, определение целей и направлений развития системы, планирование этапов развития базы данных, разработка и выпуск организационно-методических материалов.

Классификация АБД

Существует несколько видов администраторов БД, а их обязанности вполне могут отличаться от компании к компании. Вот характеристики некоторых типов АБД и занимаемых ими положений:

Оперативные (operational) АБД:

манипулируют дисковым пространством
наблюдают за текущей производительностью системы
реагируют на возникающие неисправности БД
обновляют системное ПО и ПО базы данных
контролируют структурные изменения БД
запускают процедуры резервного копирования данных
выполняют восстановление данных
создают и управляют тестовыми конфигурациями БД

Тактические (tactical) АБД:

реализуют схемы размещения информации
утверждают процедуры резервного копирования и восстановления данных;
разрабатывают и внедряют структурные элементы БД: таблицы, столбцы, размеры объектов, индексацию и т.п.;
сценарии(scripts) изменения схемы БД;
конфигурационные параметры БД
утверждают план действий в случае аварийной ситуации

Стратегические (strategic) АБД:

выбирают поставщика БД
устанавливают корпоративные стандарты данных
внедряют методы обмена данных в рамках предприятия
определяют корпоративную стратегию резервирования и восстановления данных
устанавливают корпоративный подход к ликвидации последствий аварии и обеспечению доступности данных

Старшие (senior) АБД:

досконально знают свой персонал

пользуются высоким спросом

могут написать скрипт, который освободит их из запертого сундука, брошенного в океан, и чрезвычайно гордятся своими произведениями

тратят уйму времени на подготовку младших АБД

очень ценятся руководством и получают бешеные деньги

Младшие (junior) АБД:

мечтают стать старшим АБД

не слишком сильны в написании скриптов

имеют большую склонность к использованию средств управления БД

тоже неплохо получают

Прикладные (application) АБД:

в курсе информационных нужд компании

помогают в разработке прикладных задач

отвечают за разработку схемы и ее изменения

вместе с системным АБД обеспечивают должный уровень резервирования/ восстановления данных

занимаются построением тестовых БД

Системные (system) АБД:

отвечают за все необходимое для резервирования и восстановления данных

контролируют производительность системы в целом

осуществляют поиск и устранение неисправностей

в курсе нынешних и будущих потребностей БД в плане емкости

в курсе текущего состояния и нужд БД

Наемные (contract) АБД :

приглашаются под конкретную задачу или в качестве консультантов

передают персоналу необходимые знания

фиксируют свои действия!

должны прекрасно разбираться в соответствующей области

хороши в качестве временного персонала, для оценки проекта или системы

Администраторы-руководители:

проводят еженедельные совещания

определяют перечень первоочередных задач

устанавливают и оглашают официальный курс и стратегию

утверждают и корректируют должностные инструкции и список обязанностей

следят за наличием соответствующей документации

1.2 Обязанности, связи и средства администратора современных систем управления базами данных

Поскольку система баз данных может быть весьма большой и может иметь много пользователей, должно существовать лицо или группа лиц, управляющих этой системой. Такое лицо называется администратором базы данных (АБД).

В любой базе данных должен быть хотя бы один человек, выполняющий административные обязанности; если база данных большая, эти обязанности могут быть распределены между несколькими администраторами.

В обязанности администратора могут входить:

- инсталляция и обновление версий сервера и прикладных инструментов
- распределение дисковой памяти и планирование будущих требований системы к памяти
- создание первичных структур памяти в базе данных (табличных пространств) по мере проектирования приложений разработчиками приложений
- создание первичных объектов (таблиц, представлений, индексов) по мере проектирования приложений разработчиками
- модификация структуры базы данных в соответствии с потребностями приложений
- зачисление пользователей и поддержание защиты системы
- соблюдение лицензионного соглашения
- управление и отслеживание доступа пользователей к базе данных
- отслеживание и оптимизация производительности базы данных
- планирование резервного копирования и восстановления
- поддержание архивных данных на устройствах хранения информации
- осуществление резервного копирования и восстановления
- обращение в корпорацию за техническим сопровождением

В некоторых случаях база данных должна также иметь одного или нескольких сотрудников службы безопасности. Сотрудник службы безопасности главным образом отвечает за регистрацию новых пользователей, управление и отслеживание доступа пользователей к базе данных, и защиту базы данных.

Разработчики приложений

В обязанности разработчика приложений входит:

- проектирование и разработка приложений базы данных
- проектирование структуры базы данных в соответствии с требованиями приложений
- оценка требований памяти для приложения
- формулирование модификаций структуры базы данных для приложения
- передача вышеупомянутой информации администратору базы данных
- настройка приложения в процессе его разработки
- установка мер по защите приложения в процессе его разработки

В процессе своей деятельности администратор базы данных взаимодействует с другими категориями пользователей банка данных, а также и с «внешними» специалистами, не являющимися пользователями базы данных.

Прежде всего, если банк данных создается для информационного обслуживания какого-либо предприятия или организации, то необходимы контакты с администрацией этой организации. Как указывалось выше, внедрение БД приводит к большим изменениям не только системы обработки

данных, но и всей системы управления организацией. Естественно, что такие большие проекты не могут быть выполнены без активного участия и поддержки руководителей организации. Руководство организации должно быть ознакомлено с возможностями, предоставляемыми базой данных, проинформировано об их преимуществах и недостатках, а также проблемах, вызываемых созданием и функционированием базы данных.

Так как база данных является динамическим информационным отображением предметной области, то желательно, чтобы администратор базы данных в свою очередь был своевременно информирован о перспективах развития объекта, для которого создается информационная система.

Руководством организации и администратором базы данных должны быть согласованы цели, основные направления и сроки создания БД и его развития, очередность подключения пользователей.

Очень тесная связь у АБД на всех этапах жизненного цикла базы данных наблюдается с конечными пользователями. Это взаимодействие начинается на начальных стадиях проектирования системы, когда изучаются потребности пользователей, уточняются особенности предметной области, и постоянно поддерживается как на протяжении процесса проектирования, так и функционирования системы.

Следует отметить, что в последнее время наблюдается активное перераспределение функций между конечными пользователями и администраторами банка данных. Это, прежде всего, связано с развитием языковых и программных средств, ориентированных на конечных пользователей. Сюда относятся простые и одновременно мощные языки запросов, а также средства автоматизации проектирования.

Если банк данных функционирует в составе какой-либо включающей его автоматизированной информационной системы (например, в АСУ), то АБД должен работать в контакте со специалистами по обработке данных в этой системе.

Администраторы базы данных взаимодействуют и с внешними по отношению к нему группами специалистов и, прежде всего, поставщиками СУБД и ППП, администраторами других баз данных.

базы данных часто создаются специализированными проектными коллективами на основе договора на разработку информационной системы в целом или базой данных как самостоятельного объекта проектирования. В этом случае служба администрации базы данных должна создаваться как в организации-разработчике, так и в организации-заказчике.

На эффективность работы базы данных оказывают влияние множество внешних и внутренних факторов. Возрастание сложности и масштабов базы данных, высокая «цена» неправильных или запоздалых решений по администрированию БД, высокие требования к квалификации специалистов делают актуальной задачу использования развитых средствах автоматизированного (или даже автоматического) администрирования базы данных.

Средства администрирования включены в состав всех СУБД. Особенно развиты эти средства в корпоративных СУБД. Кроме того, появился целый класс специализированного программного обеспечения: средства DBA (DataBase Administration – администрирование базы данных).
Типичные функции средств DBA представлены в таблице 1).

Мониторинг работы БД, реакция на нештатные ситуации	Наблюдение за объектами БД, анализ, сопоставление характеристик	Оптимизация хранения данных, оптимизация работы сервера	Сопровождение БД, файлов, табличных пространств, откатных сегментов
Слежение за использованием ресурсов, выдача статистики	Планирование необходимых вычислительных мощностей	Анализ свободного пространства, устранение дефрагментации	Перенос таблицы на новое пространство, в другую СУБД, на другой компьютер
Обнаружение и исправление возникающих неполадок	Задание пороговых значений для слежения за нужными объектами	Наблюдение за параметрами, влияющими на производительность БД	Перенос содержимого базы данных в другую СУБД

Таблица 1. - Типичные функции средств DBA

2.Администрирование базы данных

2.1 Управление данными в базах данных

Непосредственное управление данными во внешней памяти. Эта функция включает обеспечение необходимых структур внешней памяти как для хранения непосредственных данных, входящих в БД, так и для служебных целей, например, для убыстрения доступа к данным в некоторых случаях (обычно для этого используются индексы). В некоторых реализациях СУБД активно используются возможности существующих файловых систем, в других работа производится вплоть до уровня устройств внешней памяти. Но подчеркнем, что в развитых СУБД пользователи в любом случае не обязаны знать, использует ли СУБД файловую систему, а если использует, то как организованы в ней файлы. В частности, СУБД поддерживает собственную

систему именования объектов БД (это очень важно, поскольку имена объектов базы данных соответствуют именам объектов предметной области).

Существует множество различных способов организации внешней памяти баз данных. Как и все решения, принимаемые при организации баз данных, конкретные методы организации внешней памяти необходимо выбирать в тесной связи со всеми остальными решениями.

Управление буферами оперативной памяти. СУБД обычно работают с БД значительного размера; по крайней мере этот размер обычно существенно превышает доступный объем оперативной памяти. Понятно, если при обращении к любому элементу данных будет производиться обмен с внешней памятью, то вся система будет работать со скоростью устройства внешней памяти. Единственным же способом реального увеличения этой скорости является буферизация данных в оперативной памяти. И даже если операционная система производит общесистемную буферизацию (как в случае ОС UNIX), этого недостаточно для целей СУБД, которая располагает гораздо большей информацией о полезности буферизации той или иной части БД. Поэтому в развитых СУБД поддерживается собственный набор буферов оперативной памяти с собственной дисциплиной замены буферов. При управлении буферами основной памяти приходится разрабатывать и применять согласованные алгоритмы буферизации, журнализации и синхронизации. Заметим, что существует отдельное направление СУБД, которые ориентированы на постоянное присутствие в оперативной памяти всей БД. Это направление основывается на предположении, что в предвидимом будущем объем оперативной памяти компьютеров сможет быть настолько велик, что позволит не беспокоиться о буферизации. Пока эти работы находятся в стадии исследований.

Управление транзакциями. Транзакция – это последовательность операций над БД, рассматриваемых СУБД как единое целое. Либо транзакция успешно выполняется, и СУБД фиксирует (COMMIT) изменения БД, произведенные ею, во внешней памяти, либо ни одно из этих изменений никак не отражается в состоянии БД. Понятие транзакции необходимо для поддержания логической целостности БД.

Таким образом, поддержание механизма транзакций – обязательное условие даже однопользовательских СУБД (если, конечно, такая система заслуживает названия СУБД). Но понятие транзакции гораздо существеннее во многопользовательских СУБД. То свойство, что каждая транзакция начинается при целостном состоянии БД и оставляет это состояние целостным после своего завершения, делает очень удобным использование понятия транзакции как единицы активности пользователя по отношению к БД. При соответствующем управлении параллельно выполняющимися транзакциями со стороны СУБД каждый пользователь может в принципе ощущать себя единственным пользователем СУБД (на самом деле, это несколько идеализированное представление, поскольку пользователи

многопользовательских СУБД порой могут ощутить присутствие своих коллег).

С управлением транзакциями в многопользовательской СУБД связаны важные понятия сериализации транзакций и сериального плана выполнения смеси транзакций. Под сериализаций параллельно выполняющихся транзакций понимается такой порядок планирования их работы, при котором суммарный эффект смеси транзакций эквивлентен эффекту их некоторого последовательного выполнения. Сериальный план выполнения смеси транзакций – это такой способ их совместного выполнения, который приводит к сериализации транзакций. Понятно, что если удается добиться действительно сериального выполнения смеси транзакций, то для каждого пользователя, по инициативе которого образована транзакция, присутствие других транзакций будет незаметно (если не считать некоторого замедления работы для каждого пользователя по сравнению с однопользовательским режимом).

Существует несколько базовых алгоритмов сериализации транзакций. В централизованных СУБД наиболее распространены алгоритмы, основанные на синхронизационных захватах объектов БД. При использовании любого алгоритма сериализации возможны ситуации конфликтов между двумя или более транзакциями по доступу к объектам БД. В этом случае для поддержания сериализации необходимо выполнить откат (ликвидировать все изменения, произведенные в БД) одной или более транзакций. Это один из случаев, когда пользователь многопользовательской СУБД может реально (и достаточно неприятно) ощутить присутствие в системе транзакций других пользователей.

Журнализация. Одно из основных требований к СУБД – надежное хранение данных во внешней памяти. Под надежностью хранения понимается то, что СУБД должна быть в состоянии восстановить последнее согласованное состояние БД после любого аппаратного или программного сбоя. Обычно рассматриваются два возможных вида аппаратных сбоев: так называемые мягкие сбои, которые можно трактовать как внезапную остановку работы компьютера (например, аварийное выключение питания), и жесткие сбои, характеризуемые потерей информации на носителях внешней памяти. Примерами программных сбоев могут быть аварийное завершение работы СУБД (из-за ошибки в программе или некоторого аппаратного сбоя) или аварийное завершение пользовательской программы, в результате чего некоторая транзакция остается незавершенной. Первую ситуацию можно рассматривать как особый вид мягкого аппаратного сбоя; при возникновении последней требуется ликвидировать последствия только одной транзакции.
Но в любом случае для восстановления БД нужно располагать некоторой дополнительной информацией. Другими словами, поддержание надежного хранения данных в БД требует избыточности хранения данных, причем та их часть, которая используется для восстановления, должна храниться особо

надежно. Наиболее распространенный метод поддержания такой избыточной информации – ведение журнала изменений БД.

Журнал – это особая часть БД, недоступная пользователям СУБД и поддерживаемая особо тщательно (иногда поддерживаются две копии журнала, располагаемые на разных физических дисках), в которую поступают записи обо всех изменениях основной части БД. В разных СУБД изменения БД журнализуются на разных уровнях: иногда запись в журнале соответствует некоторой логической операции изменения БД (например, операции удаления строки из таблицы реляционной БД), а порой запись соответствует минимальной внутренней операции модификации страницы внешней памяти. В некоторых системах одновременно используются оба подхода.

Во всех случаях придерживаются стратегии «упреждающей» записи в журнал (так называемого протокола Write Ahead Log – WAL). Грубо говоря, эта стратегия заключается в том, что запись об изменении любого объекта БД должна попасть во внешнюю память журнала раньше, чем измененный объект попадет во внешнюю память основной части БД. Известно, если в СУБД корректно соблюдается протокол WAL, то с помощью журнала можно решить все проблемы восстановления БД после любого сбоя.

Самая простая ситуация восстановления – индивидуальный откат транзакции. Строго говоря, для этого не требуется общесистемный журнал изменений БД. Достаточно для каждой транзакции поддерживать локальный журнал операций модификации БД, выполненных в этой транзакции, и производить откат транзакции выполнением обратных операций, следуя от конца локального журнала. В некоторых СУБД так и делают, но в большинстве систем локальные журналы не поддерживают, а индивидуальный откат транзакции выполняют по общесистемному журналу, для чего все записи от одной транзакции связывают обратным списком (от конца к началу).

При мягком сбое во внешней памяти основной части БД могут находиться объекты, модифицированные транзакциями, не закончившимися к моменту сбоя, и могут отсутствовать объекты, модифицированные транзакциями, которые к моменту сбоя успешно завершились (по причине использования буферов оперативной памяти, содержимое которых при мягком сбое пропадает). При соблюдении протокола WAL во внешней памяти журнала должны гарантированно находиться записи, относящиеся к операциям модификации обоих видов объектов. Целью процесса восстановления после мягкого сбоя является состояние внешней памяти основной части БД, которое возникло бы при фиксации во внешней памяти изменений всех завершившихся транзакций и которое не содержало бы никаких следов незаконченных транзакций. Чтобы этого добиться, сначала производят откат незавершенных транзакций (undo), а потом повторно воспроизводят (redo) те операции завершенных транзакций, результаты которых не отображены во внешней памяти. Этот процесс содержит много

тонкостей, связанных с общей организацией управления буферами и журналом. Более подробно мы рассмотрим это в соответствующей лекции.
Для восстановления БД после жесткого сбоя используют журнал и архивную копию БД. Грубо говоря, архивная копия – это полная копия БД к моменту начала заполнения журнала (имеется много вариантов более гибкой трактовки смысла архивной копии). Конечно, для нормального восстановления БД после жесткого сбоя необходимо, чтобы журнал не пропал. Как уже отмечалось, к сохранности журнала во внешней памяти в СУБД предъявляются особо повышенные требования. Тогда восстановление БД состоит в том, что исходя из архивной копии по журналу воспроизводится работа всех транзакций, которые закончились к моменту сбоя. В принципе можно даже воспроизвести работу незавершенных транзакций и продолжить их работу после конца восстановления. Однако в реальных системах это обычно не делается, поскольку процесс восстановления после жесткого сбоя является достаточно длительным.

2.3 Управление безопасностью в СУБД

Системы управления базами данных стали основным инструментом, обеспечивающим хранение больших массивов информации. Современные информационные приложения опираются, как уже говорилось, в первую очередь, на многопользовательские СУБД. В этой связи пристальное внимание в настоящее время уделяется проблемам обеспечения информационной безопасности, которая определяет степень безопасности организации, учреждения в целом.

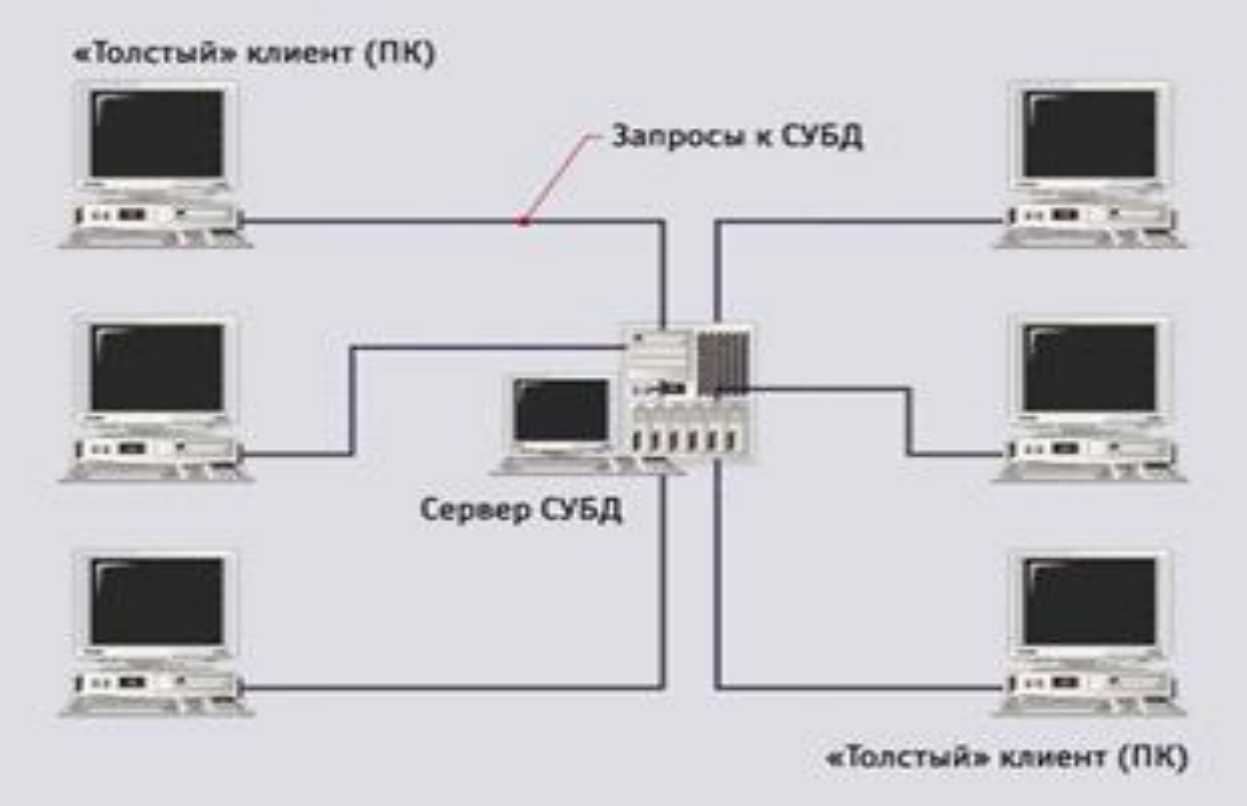

Рис.2 - Архитектура «клиент-сервер»

Под информационной безопасностью понимают защищенность информации от случайных и преднамеренных воздействий естественного или искусственного характера, чреватых нанесением ущерба владельцам или пользователям информации.

Защите подлежат не только данные; в базе данных нарушения в защите могут повлиять на другие части системы, что повлечет за собой разрушение и базы данных. Поэтому защита баз данных охватывает и оборудование, и программное обеспечение, и персонал, и, собственно, данные.

Таким образом, защита баз данных предусматривает предотвращение любых преднамеренных и непреднамеренных угроз с использованием компьютерных и некомпьютерных средств контроля. Следует также защищать: современные информационные системы; глобальную связанность (выход в Internet); разнородность (различные платформы); технологию «клиент-сервер». (рис.2)

Надо отметить, что, разрабатывая систему информационной безопасности, надо помнить, что только защищая все составные части системы, можно достичь желаемого результата.

Рассмотрим основные программно-технические меры, применение которых позволит решить некоторые из вышеперечисленных проблем: аутентификация и идентичность; управление доступом; поддержка целостности; протоколирование и аудит; защита коммуникаций между клиентом и сервером; отражение угроз, специфичных для СУБД.

Аутентификация и идентичность. Проверка подлинности пользователя приложений базы данных чаще всего осуществляется либо через соответствующие механизмы операционной системы, либо через определенный SQL-оператор: пользователь идентифицируется своим именем, а средством аутентификации служит пароль. Авторизация –предоставление прав (привилегий), позволяющих владельцу иметь законный доступ к объектам базы данных. Аутентификация – механизм определения того, является ли пользователь тем, за кого он себя выдает. Эта процедура позволяет организовать контролируемый доступ к информационной системе (пользователь – идентификатор – пароль). Пароль – наиболее распространенный метод аутентификации, но он не дает абсолютной гарантии, что пользователь является именно тем, за кого себя выдает. При использовании такого подхода создаются значительные сложности для повторных проверок и исключает подобные проверки перед каждой транзакцией. Средства аутентификации на основе личных карточек или эквивалентного механизма дали бы приложению большую свободу в реализации контроля за подлинностью пользователей.

Управление доступом. После получения права доступа к СУБД пользователь автоматически получает привилегии, связанные с его идентификатором. Это может относиться к процедурам доступа к объектам базы данных, к операциям над данными. Для основных объектов базы данных

могут быть построены таблицы, в которых указывается набор действий, доступных каждому пользователю системы.

Каждому возможному действию над данными таблицы ставится в соответствие двоичное значения, общий результат возможных операция получается путем суммирования набранных пользователем значений.

Привилегии в СУБД могут быть разделены на две категории: привилегии безопасности и привилегии доступа. Привилегии безопасности позволяют выполнять административные действия, привилегии доступа определяют права доступа субъектов к определенным объектам.

До выполнения процедуры присваивания привилегий их необходимо создать. Привилегии можно подразделить в соответствии с видами объектов, к которым они относятся: таблицы и представления, процедуры, базы данных сервер базы данных. Применительно к таблицам могут быть определены следующие права доступа: право на выборку, удаление, обновление, добавление, право на использование внешних ключей, ссылающихся на данную таблицу. По умолчанию пользователь не имеет никаких прав доступа ни таблицам, ни к представлениям. По отношению к процедурам можно предоставить право на их выполнение, однако при этом не указываются привилегии на право доступа к объектам, обрабатываемым процедурами. Это позволяет выделять неконтролируемый доступ для выполнения строго определенных операций над данными. По отношению к базе данных выделяемые права на самом деле являются запретительными: ограничение на число операций ввода/вывода строк, число строк, возвращаемым одним запросом.

Оператор позволяет реализовать следующие виды ограничений доступа:

- операционные ограничения (за счет прав доступа операторов выборки, вставки, удаления и обновления, применяемые ко всем или отдельным столбцам таблицы);
- ограничения по значениям (за счет механизма представлений);
- ограничения на ресурсы (за счет привилегий к базам данных).

СУБД предоставляет специфическое средство управление доступом – представления, которые позволяют делать видимыми только отдельные столбцы базовых таблиц. В результате выполнения запроса к представлению, а не к таблице может быть возвращена таблица из нуля строк, а не код ответа, свидетельствующий о нарушении прав доступа. Это важно, поскольку этот ответ лишает возможности поиска ответа другим путем, например, через анализ кодов, ответов, возвращаемых после обработки SQL-запросов.

Управление доступом базируется на реализации следующего минимального набора действий: произвольное управление доступом; обеспечение безопасности повторного использования объектов; использование меток безопасности; принудительное управление доступом.

Произвольное управление доступом – метод ограничения доступа к объектам, основанный на учете личности субъекта или групп, в которую субъект входит. Группа – это именованная совокупность пользователей;

объединение субъектов в группы облегчает процесс администрирования данных, и строится на основе формальной структуры организации. Эта технология обеспечивает владельцу объекта (представления, сервера базы данных, процедуры, таблицы) передачу по своему усмотрению привилегий другому лицу. Этим лицом в данной ситуации может выступать субъект-пользователь, группа пользователей и такой возможный носитель привилегий как роль.

Привилегии распределяются в зависимости от статуса пользователей (администратор сервера базы данных и база данных, прочие пользователи). Особенно важным следует считать поддержание уровня защиты привилегий администратора сервера базы данных, т.к. компрометация его пароля может привести к серьезным проблемам для всей хранящейся информации.

Главное достоинство применения произвольного управления доступом – гибкость, однако такие сопутствующие характеристики как рассредоточенность управления и сложность централизованного контроля создают немало проблем для обеспечения безопасности данных.

В качестве дополнения к средствам управления доступа можно отнести безопасность повторного использования объектов, которая должна гарантироваться для областей оперативной памяти, в частности, для буферов с образами экрана, расшифрованными паролем, для магнитных носителей. Следует обратить внимание и на обеспечение безопасности повторного использования субъектов. Это означает лишение прав для входа в информационную систему всех пользователей, покинувших организацию.

Специалисты по управлению безопасностью информации считают достаточным для большинства коммерческих приложений использование средств произвольного управления доступом. Тем не менее, останется нерешенной одна важная проблема – слежение за передачей информации.

Средства произвольного управления доступом не могут помешать авторизованному пользователю законным путем получить конфиденциальную информацию и сделать ее доступной для других пользователей. Это происходит по той причине, что при произвольном управлении доступом привилегии существуют изолированно от данных. Для решения этой проблемы применяют подход, позволяющий осуществить разделение данных по уровням секретности и категориям доступа, называемый метками безопасности.

Метка безопасности состоит из двух частей: уровня секретности и списка категорий. Первая составляющая зависит от приложения и в стандартном варианте может выглядеть как спектр значений от «совершенно секретно» до «несекретно». Вторая составляющая позволяет описать предметную область, разделяя информацию «по отсекам», что способствует лучшей защищенности. Метки безопасности строк таблица неявно добавляются к каждой строке реляционного отношения.

Каждый пользователь также получает и собственную метку безопасности, которая определяет степень его благонадежности. Пользователь

получает доступ к данным, если степень его благонадежности соответствует требованиям соответствующей метки безопасности, а именно: уровень секретности пользователя должен быть не ниже уровня секретности данных; набор категорий, заданных в метке безопасности данных, должен целиком находиться в метке безопасности пользователя.

Механизм меток безопасности не отменяет, а дополняет произвольное управление доступом: пользователи по-прежнему могут оперировать с таблицами только в рамках своих привилегий, но получать только часть данных столбец в метками безопасности не включается в результирующую таблицу). Основная проблема, имеющая место при использовании меток безопасности, поддержание их целостности. Это означает, что все объекты и субъекты должны быть помечены и при любых операциях с данными метки должны оставаться правильными. При добавлении или изменении строк метки, как правило, наследуют метки безопасности пользователя, инициировавшего операцию. Таким образом, даже если авторизованный пользователь перепишет секретную информацию в общедоступную таблицу, менее благонадежные пользователи не смогут ее прочитать.

Принцип принудительного управления доступом основан на сопоставлении меток безопасности субъекта и объекта. Для чтения информации из объекта необходимо доминирование метки субъекта над меткой объекта. При выполнении операции записи информации в объект необходимо доминирование метки безопасности объекта над меткой субъекта. Этот способ управления доступом называется принудительным, т.к. не зависит от воли субъектов. Он нашел применение в СУБД, отличающихся повышенными мерами безопасности.

Поддержка целостности. Обеспечение целостности данных не менее важная задача, чем управление доступом. Главными врагами баз данных являются ошибки оборудования, администраторов, прикладных программ и пользователей, а не злоумышленников. С точки зрения пользователей СУБД, основными средствами поддержания целостности данных являются: ограничения; правила.

Ограничения могут поддерживаться непосредственно в рамках реляционной модели данных, а могут задаваться в процессе создания таблицы. Табличные ограничения могут относиться к группе столбцов, отдельным атрибутам. Ссылочные ограничения отвечают за поддержание целостности связей между таблицами. Ограничения накладываются владельцем таблицы и влияют на результат последующих операций с данными. Ограничения являются статическим элементом поддержания целостности, т.к. они или разрешают выполнять действие или нет.

Правила позволяют вызывать выполнение заданных процедур при определенных изменениях базы данных. Правила ассоциируются с таблицами и срабатывают при изменении этих таблиц. В отличие от ограничений, которые обеспечивают контроль относительно простых условий, правила позволяют проверять и поддерживать соотношения любой сложности между

элементами данных в базе. В контексте информационной безопасности необходимо отметить, что создание правила, ассоциированного с таблицей, может реализовать только владелец этой таблицы. Пользователь, действия которого вызывают срабатывание правила, должен обладать лишь необходимыми правами доступа к таблице. Тем самым правила неявно расширяют привилегии пользователя. Подобные расширения должны контролироваться административно, так как даже незначительное изменение правила может кардинально повлиять на защищенность данных. Существует явное предостережение при использовании правил как инструмента информационной безопасности: ошибка в сложной системе правил чревата непредсказуемыми последствиями для всей базы данных.

Протоколирование и аудит. Аудит – проверка того, все ли предусмотренные средства управления задействованы и соответствуют уровню защищенности установленным требованиям. Такая мера как протоколирование и аудит состоит в следующем: обнаружение необычных и подозрительных действий пользователей и идентификация лиц, совершивших эти действия; оценка возможных последствий состоявшегося нарушения; оказание помощи; организация пассивной защиты информации от нелегальных действий пользователя: поддержка точности вводимых данных; поддержка документации в активном виде; корректное тестирование пользователей.

Рекомендуется при выполнении организации протоколирования фиксировать факты передачи привилегий и подключения к той или иной базе данных.

Средства поддержки доступности. Следующим вопросом в рассмотрении проблемы обеспечения информационной безопасности является анализ средств поддержания высокой готовности. Если речь идет о СУБД, то необходимо в архитектуре аппаратно-программного комплекса иметь средства, обеспечивающие нейтрализацию аппаратных отказов и восстановление после ошибок обслуживающего персонала или прикладных программ.

Сохранение информации базы данных на диски, помещаемые затем в безопасное место, уже длительное время активно применяется для информационных систем. При архивировании сохраняются пространства базы данных и многочисленная сопутствующая информация, необходимая для последующего восстановления. Резервное копирование – периодически выполняемая процедура получения копии базы данных и ее журнала изменений на носителе, сохраняемом отдельно от системы. Надо помнить, что архив отражает состояние базы данных на время начала архивирования. Резервные копирования логических журналов транзакций сохраняет файлы журналов; интерпретация последних позволяет восстановить базу данных до состояния, более позднего, чем момент последнего архивирования. На основании полученной информации из архива и/или резервной копии может быть осуществлено восстановление как архивной информации (физическое

восстановление), так и более свежие состояние базы данных (логическое восстановление). Можно перечислить возможности службы восстановления на примере СУБД Informix: архивировании в горячем режиме, т.е. без прерывания работы сервера; резервное копирование журналов транзакций; сохранение на удаленных устройствах; сохранение по заранее определенному расписанию без участия оператора; сжатие и шифрование информации. Контрольные точки – момент синхронизации между состоянием базы данных и журнала транзакция. В это время принудительно выгружаются содержимое буфера оперативной памяти на устройства вторичной памяти.

Рассмотренные выше средства сохранения могут обеспечить восстановление с минимальными потерями пользовательской информации, однако работа сервера базы данных будет на некоторое время прервана.

Следующий механизм, обеспечивающий высокий уровень отказоустойчивости – технология тиражирования данных. Тиражирование данных – это асинхронный перенос изменений объектов исходной базы данных в базы, принадлежащие различным узлам распределенной системы. В конфигурации серверов базы данных выделяют один основной и ряд вторичных. В обычном режиме работы основной сервер выполняет чтение и обновление данных, обеспечивает перенос зафиксированных изменений на вторичные серверы. В случае отказа основного сервера его функции автоматически (вручную) переводятся на вторичный сервер в режим работы “чтение + запись”. После восстановления функций основного сервера ему может быть присвоен статус вторичного, а вторичному делегированы все полномочия основного (при этом обеспечивается непрерывная доступность данных). Процедура тиражирования осуществляется либо в синхронном, либо в асинхронном режиме. Благодаря первому осуществляется гарантированность полной согласованности данных, т.е. на вторичном сервере будут зафиксированы все транзакции, выполненные на основном. Асинхронный режим улучшает рабочие характеристики системы, не всегда обеспечивая полную согласованность (стоит заметить, что далеко не во всех задачах требуется синхронизация фиксации; достаточно поддерживать тождественность данных лишь в критические моменты времени).

Защита коммуникаций между клиентом и сервером. Проблема защиты коммуникаций между клиентом и сервером в информационных системах не является специфичной для СУБД. Для обеспечения защиты информации выделяется сервис безопасности, в функции которого входит аутентификация, шифрование и авторизация. Эти вопросы были рассмотрены выше.

Отражение угроз, специфичных для СУБД. Однако главный источник угроз для СУБД лежит в самой природе баз данных. Наличие специфического непроцедурного языка SQL, процедур и механизм правил – все это обеспечивает потенциальному злоумышленнику средства для получения информации, на которую он не имеет полномочий. Нередко нужную, но недоступную по статусу информацию можно получить путем логического вывода. Например, используя операцию вставки, а не выбора (на которую прав

нет), анализировать коды завершения SQL-операторов, и получать информацию о наборе первичных ключей. Для борьбы с подобными угрозами используется механизм размножения строк для СУБД, поддерживающий метки безопасности. Суть этого метода состоит в том, чтобы в состав первичного ключа добавлять метку безопасности, что обеспечивает одновременное хранение в базовой таблице несколько экземпляров строк с одинаковыми значениями важных ключей.

Агрегирование – метод получения новой информации путем комбинирования данных, добытых легальным путем из различных таблиц базы данных. Бороться с агрегированием можно за счет тщательного проектирования модели данных и максимального ограничения доступа пользователя к информации.

Таким образом, можно определить некоторые стратегии в области безопасности: минимум привилегий; разделение обязанностей; эшелонированная оборона (сервер базы данных, средства защиты СУБД и операционной системы); разнообразие средств защиты; невозможность перехода в небезопасное состояние; всеобщая поддержка мер безопасности; «человеческий фактор».

Заключение

На основании проведенного исследования «Администрирование баз данных» можно сделать следующие выводы.

Администрирование базами данных предусматривает выполнение функций, направленных на обеспечение надежного и эффективного функционирования системы баз данных, адекватности содержания базы данных информационным потребностям пользователей, отображения в базе данных актуального состояния предметной области.

Администратор БД отвечает за целостность информационных ресурсов компании. На нем лежит ответственность по созданию, обновлению и сохранности связанных между собой резервных копий файлов, исходя из задач предприятия. Этот человек должен в мельчайших подробностях знать существующие механизмы восстановления программного обеспечения БД.
Возможны ситуации, при которых администратору БД потребуется на основе логических прикладных моделей создавать элементы физической схемы, а также поддерживать связь пользователей с системой и обеспечивать соответствующий уровень информационной безопасности, следя за тем, чтобы доступ к данным имели только те люди, которые в нем нуждаются.
Администратор БД должен уметь определять узкие места системы, ограничивающие ее производительность, настраивать SQL и программное обеспечение СУБД и обладать знаниями, необходимыми для решения вопросов оптимизации быстродействия БД.

Администратор базы данных (АБД) должен координировать действия по сбору сведений, проектированию и эксплуатации базы данных, а также по

обеспечению защиты данных. Он обязан учитывать текущие и перспективные информационные требования предметной области, что является одной из главных задач.

Правильная реализация функций администрирования базы данных существенно улучшает контроль и управление ресурсами данных предметной области. С этой точки зрения функции АБД являются больше управляющими, нежели техническими. Принципы работы АБД и его функции определяются подходом к данным как к ресурсам организации, поэтому решение проблем, связанных с администрированием начинается с установления общих принципов эксплуатации СУБД.

Важная задача АБД состоит в устранении противоречий между различными направлениями деятельности организации по создании концептуальной, а затем и логической схемы данных предметной области. Кроме определения данных и прав доступа, от АБД может потребоваться разработка процедур и руководств по ведению данных. В процессе сбора информации АБД должен уметь пользоваться своей властью и влиянием, обладать определенным стажем работы и хорошо разбираться в обстановке в компании. АБД необходимо установить эффективную взаимосвязь со всеми группами сотрудников, которым приходиться обращаться с базой данных.

Таким образом, можно сделать определенные обобщения.

Администратор базы данных – это: управляющий данными, а не хозяин; системный программист определенного профиля, а также эксперт высшего уровня, обеспечивающий службу эксплуатации решениями по процедурам и регламентам работы; лицо, принимающее окончательное решение в своей области, и человек, обладающий способностями к общению, совместному планированию и компромиссам.

Надежность и достоверность – это ключевые понятия в деятельности администратора базы данных. Он должен уметь вести тщательное документирование всех действий по управлению базой данных.

Глоссарий

№№ п/п	Понятие	Содержание
1	2	3
1	База данных	совокупность связанных данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования, независимая от прикладных программ. База данных является информационной моделью предметной области. Обращение к базам данных осуществляется с помощью системы управления базами данных (СУБД
2	Документографическая база данных	база данных, в запись которой отражает конкретный документ, содержит его библиографическое описание и, возможно, иную информацию о нем
3	Информационный продукт	документированная информация, подготовленная в соответствии с потребностями пользователей и представленная в форме товара. Информационными продуктами являются программные продукты, базы и банки данных и другая информация
4	Локальная база данных	база данных, размещенная на одном или нескольких носителях на одном компьютере
5	Объектно-ориентированная база данных	база данных, в которой данные оформлены в виде моделей объектов, включающих прикладные программы, которые управляются внешними событиями
6	Распределенная база данных	совокупность баз данных, физически распределенная по взаимосвязанным ресурсам вычислительной сети и доступная для совместного использования

7	Реляционная база данных	база данных, построенная на основе реляционной модели. В реляционной базе каждый объект задается записью (строкой) в таблице. Реляционная база создается и затем управляется с помощью реляционной системы управления базами данных
8	Система управления базами данных (СУБД)	комплекс программных и лингвистических средств общего или специального назначения, реализующий поддержку создания баз данных, централизованного управления и организации доступа к ним различных пользователей в условиях принятой технологии обработки данных
9	Система управления распределенными базами данных	система управления базами данных, содержимое которых располагается в нескольких абонентских системах информационной сети. В СУРБД используется комбинация централизованного и локального способов хранения данных
10	Структура базы данных	принцип или порядок организации записей в базе данных и связей между ними

Источники:

1.Веретенникова Е.Г., Патрушина С.М., Савельева Н.Г. Информатика: Учебное пособие. Серия «Учебный курс», –М., 2002.
2.Гуде С.В., Ревин С.Б. Информационные системы. Учебное пособие. –М., 2002.
3.Дунаев С. Доступ к базам данных и техника работы в сети. Практические приемы современного программирования. – М., 2005.
4.Информатика: Учебник/Каймин В.А., 2-е изд. перераб. и доп. – М: Инфра-М., 2002.
5.Информатика: Учебник/Под ред. Н.В.Макаровой, 3-е изд., перераб. и доп. – М.: Финансы и статистика, 2001.
6.Информатика: Учебник для вузов/Острейковский В.А., М: Высшая школа, 2001.
7.Информатика: Учебник для вузов/Козырев А.А. – СПб., 2002.
8.Мейер Д. Теория реляционных баз данных: пер. с англ. – М., 2005.
9.Ревунков Г.И., Самохвалов Э.Н., Чистов В.В. Базы и банки данных и знаний. Учебник для вузов//Под ред. В.Н.Четверикова. – М., 2003.
10.Фаронов В.В., Шумаков П.В. Руководство разработчика баз данных. – М.: Нолидж, 2000.
11.Фундаментальные основы информатики: социальная информатика.: Учебное пособие для вузов / Колин К.К. – М.: Академ.проект: Деловая книга Екатеринбург, 2000.