Содержание:

ВВЕДЕНИЕ

Коллективное использование базы данных требует административного контроля. Эти обязанности поручаются одному или нескольким сотрудникам, которые будут исполнять роль администраторов баз данных.

Некоторые полагают, что использование базы данных небольшим количеством пользователей не требует особенного обслуживания, и считают нецелесообразным выделение специальных средств на административную поддержку базы данных, аргументируя это отсутствием свободных сотрудников и нехваткой служебного времени.

Однако если этот вопрос игнорировать, так или иначе такое положение дел приведет к нежелательным последствиям. Однажды база данных может быть повреждена и не будет резервных копий. Невозможно будет решить некоторые часто возникающие задачи администрирования, требующие определенной квалификации, которой пользователи, как правило, не имеют.

Главные задачи администрирования базы данных – обеспечение надежной и

эффективной работы базы данных, адекватности содержания базы данных.

информационным потребностям пользователей, отображения в БД актуального

состояния данных.

1. ОСНОВНЫЕ ПОНЯТИЯ АДМИНИСТРИРОВАНИЯ БАЗ ДАННЫХ

- 1.

Понятие и функции администратора базы данных

Нормальная работа базы данных становится невозможной без участия специалистов, обеспечивающих создание, функционирование и развитие базы данных. Такие специалисты называются администратором базы данных (АБД). Группа таких специалистов считается составной частью базы данных.

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

АБД выполняют работы по созданию и обеспечению функционирования баз данных на протяжении всех этапов жизненного цикла системы. В составе группы администраторов баз данных можно выделить различные подгруппы в зависимости от выполняемых ими функций.

Администраторы базы данных имеют различные функции:

1. Анализируют предметную область: 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. Проектируют структуру 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 базы данных на языке обработки данных.
3. Осуществляют 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. Первоначально загружают и ведут базу данных: разработка технологии первоначальной загрузки и ведения (изменения, добавления, удаления записей) базы данных, проектирование форм ввода, создание программных модулей, подготовка исходных данных, ввод и контроль ввода.
5. Защищают данные 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.

1. Защищают данные от разрушений. Одним из способов защиты от потери данных является резервирование. Используется как при физической порче файла, так и в случае, если в базу данных внесены нежелательные необратимые изменения.
2. Обеспечивают восстановление базы данных: разработка программно-технологических средств восстановления базы данных, организация ведения системных журналов.
3. Анализируют обращение пользователей к базе данных: сбор статистики обращений пользователей к базе данных, ее хранение и анализ (кто из пользователей, к какой информации, как часто обращался, какие выполнял операции, время выполнения запросов, анализ причин безуспешных (в том числе и аварийных) обращений к базе данных.
4. Анализируют эффективность 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.
5. Работают 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.
6. Подготавливают и поддерживают системные программные средства: сбор и анализ информации о СУБД и других прикладных программ, приобретение программных средств, их установка, проверка работоспособности, поддержание системных библиотек, развитие программных средств.
7. Занимаются организационно-методической работой: выбор или создание методики проектирования БД, определение целей и направлений развития системы, планирование этапов развития базы данных, разработка и выпуск организационно-методических материалов.

1.2 Обязанности, связи и средства администратора современных СУБД

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.

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 было указано 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).

2. АДМИНИСТРИРОВАНИЕ БАЗ 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.

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.

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.

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. 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.

2.2 Управление безопасностью в СУБД

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).

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

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.

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.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Диго С.М. БАЗЫ ДАННЫХ. ПРОЕКТИРОВАНИЕ И СОЗДАНИЕ: Учебно-методический комплекс. – М.: Изд. центр ЕАОИ. 2008. – 171 с.
2. Иллюстрированный самоучитель по Microsoft Access http://www.taurion.ru/access
3. Гайдамакин Н.А. Автоматизированные информационные системы, базы и банки данных. Вводный курс: Уч. пос./ Н.А. Гайдамакин – М.: Гелиос АРВ, 2002. – 318с.