

## Содержание:

image not found or type unknown



## Введение

**Администратор** - лицо, ответственное за целостность и непротиворечивость данных в системе, безопасность системы, эффективность функционирования системы и использования ею ресурсов. СУБД (система управления баз данных) "видит" администратора как пользователя, обладающего определенным набором привилегий. Привилегии администратора дают ему возможность использовать такие команды и утилиты СУБД и иметь доступ к таким системным таблицам, которые недоступны рядовым пользователям. Как правило, СУБД предоставляют в распоряжение администратора еще и специальный инструментарий, который обеспечивает удобный интерфейс для выполнения функций администратора. Во всех СУБД различаются (хотя и называются по-разному) два уровня администрирования: системный администратор (администратор СУБД) и администраторы базы данных (БД). Одна копия программного продукта СУБД может поддерживать одновременное существование многих БД - коллекций данных и прикладных средств их обработки. Разные БД могут быть связаны с разными проектами и даже с разными организациями, поэтому у каждой БД должен быть свой администратор. Функции системного администратора относятся ко всей системе в целом, его права и привилегии распространяются на все объекты и на все субъекты в системе. Функции администратора БД относятся не только к подмножеству системных ресурсов, выделенных конкретной БД, его права и привилегии распространяются на объекты, относящиеся к данной БД, и на субъекты, имеющих к ним доступ. Руководства для разных СУБД по-разному формулируют функции администратора, но в конечном итоге они сводятся к следующим:

- инсталляция СУБД;
- управление памятью;
- управление разделением данных между пользователями;
- копирование и восстановление БД;
- управление безопасностью в системе;

- передача данных между СУБД и другими системами;
- управление производительностью.

Инсталляция СУБД является функцией только системного администратора. Данные в СУБД хранятся на внешней памяти. Администратор должен обеспечить такое выделение памяти, чтобы с одной стороны, ее было достаточно для хранения и эффективного доступа к данным, а с другой - минимальное количество выделенной памяти оставалось неиспользованным. Разделение данных между пользователями при их параллельной работе обеспечивается автоматически средствами СУБД и поддерживается средствами языка SQL. Однако при одновременной работе независимых приложений (иногда - и в рамках одного приложения) могут возникать конфликты одновременного доступа. Администратор, имея исчерпывающее представление о дисциплинах разделения, применяемых СУБД, выступает в роли консультанта прикладных программистов, сводя к минимуму взаимное блокирование приложениями друг друга. Копирование и восстановление являются необходимыми для гарантирования сохранности данных даже при полном крахе системы. Эта часть функций администратора включает в себя работу с соответствующими утилитами СУБД и с протоколами транзакций. Управление безопасностью данных защищает их от несанкционированных пользователей. Оно состоит в регистрации пользователей в системе, выделении пользователям привилегий и бюджетов. Данные, хранящиеся в БД, могут потребоваться для использования в других БД, работающих в другой инсталляции, или в приложениях, не зависящих от СУБД. Для целей переноса данных в распоряжении администрации имеются утилиты выгрузки данных в формате, пригодном для переноса и, соответственно, загрузки данных, поступивших из другой системы. Управление производительностью включает в себя три аспекта: настройку параметров функционирования самой СУБД, отдельных БД и отдельных приложений. Первое обеспечивается конфигурированием системы и использованием системных утилит. Второе - составом и структурой компонентов БД (таблиц, индексов, триггеров и т.п.), третье - выбором средств разработки и оптимизацией формулировок запросов, т.е. зависит в основном от прикладного программиста.

## **Обязанности администратора базы данных (АБД)**

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

Сотрудники службы безопасности. 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

## **Обеспечение защиты базы данных**

Защита обеспечивается с помощью средств создания, изменения и уничтожения учётных записей пользователя базы данных, а также с помощью привилегий (Grant, role), регулирующих возможность создавать и изменять объекты базы данных.

## **Обеспечение целостности базы данных**

Целостность данных определением правил проверки достоверности данных гарантирующих, что недействительные данные не попадут в ваши таблицы. При этом они активируются всякий раз, когда какой-либо вид изменение проводится в таблице, независимо от того, какая программа выполняет вставки, модификации или удаления. Этот контроль осуществляется в форме ограничений и триггеров БД. Ограничения √ это правила, применимые к таблицам во времена или после создания, распространяемые на то, как эти таблицы могут заполняться.

Ограничение целостности устанавливает правила на уровне БД, определяя набор проверок для таблиц системы. Эти проверки автоматически выполняются всякий раз, когда вызывается оператор вставки, модификации или удаления данных в таблице. Если какие-либо ограничения нарушены, операторы отменяются.

Поскольку ограничения условности проверяются на уровне БД, они выполняются независимо от того, откуда были инициированы операторы вставки, модификации или удаления. Для таблиц можно задавать следующие типы ограничений целостности:

NOT NULL

PRIMARY KEY

UNIQUE KEY

FOREIGN KEY (REFERENCES)

CHECK

## INDEX (ИНДЕКСЫ)

## TRIGGERS и PROCEDURES

## NOT NULL.

Это ограничение устанавливается для столбца, чтобы указать, что столбец должен иметь значение в каждой строке, т.е. некоторое непустое значение.

**PRIMARY KEY** (первичный ключ). Ограничение определяет столбец или группу столбцов, которую можно использовать для уникальной идентификации строки. Никакие две строки в таблице не могут иметь одинаковые значения столбцов первичного ключа. Кроме того, столбцы первичного ключа должны всегда содержать значение. Все эти условия гарантируют то, что в нашем распоряжении будет одна и только одна строка, соответствующая критериям связывания. Первичные ключи могут быть или именованные (пользователем) или неименованные. В первичных ключах не могут использоваться столбцы типа: raw, long, long raw.

**UNIQUE** (уникальный). Ограничение UNIQUE используется для определения того, что значения в столбце не должны повторяться в другой строке этой таблицы, определяет вторичный ключ для таблицы. Это столбец или группа столбцов, которые можно использовать как уникальную идентификацию строки. Никакие две строки не могут иметь одинаковые значения для столбца или столбцов ключа UNIQUE. Столбцы для ограничения UNIQUE не обязательно NOT NULL. Можно сформировать ограничение таблицы, указав, что в таблице не должна повторяться комбинация столбцов. К примеру: можно в начале объявить стандартно emp\_id number, person\_id date а под конце объявить что: unique(emp\_id, person\_id) ✓ и получится, что сочетание значений этих полей, должно быть уникальным в каждой строке.

**FOREIGN KEY** (внешний ключ). FOREIGN KEY, устанавливает отношение целостности между таблицами. Оно требует, чтобы столбец или набор столбцов в одной таблице совпадал с первичным или вторичным ключом другой таблицы. С момента создания внешнего ключа, ссылающегося на первичный ключ некой таблицы удаление таблицы будет ✓ запрещено. И обойти это ограничение можно только удалив ограничение. Внешние ключи могут быть именованные или неименованные.

**CHECK**. Ограничение CHECK определяет логику проверки, которая должна жать результат true (истина) для оператора вставки, модификации или удаления из

таблицы. Ограничение CHECK гарантирует, что значение в измененной строке удовлетворяют заданному набору проверок правильности.

ИНДЕКСЫ (INDEX). Ограничения PRIMARY KEY и UNIQUE автоматически создают индексы на столбцах, для которых они определены, если ограничение активизируется при создании. Если индекс уже существует на столбцах, которые составляют ограничение PRIMARY KEY и UNIQUE.

TRIGGERS(Триггеры) √ с программный элемент хранимый в БД выполняемый автоматически, в определенных ситуациях, не имеющих входных или выходных параметров, что в конечном итоге и является причиной невозможности вызвать его явно, непосредственно, его вызывает только сама база данных. Выходные данные триггера должны быть также применимы к БД, а не возвращены вызывающей программе или отображены на экране. Процедуры и функции расписаны в вопросе и объекты схемы

Таким образом, целостность базы данных может быть рассмотрена на трех уровнях:

На уровне типа данных (т.е. соответствия типов данных)

На уровне ключей (к примеру, соответствие первичных и внешних ключей)

На уровне триггеров, процедур и/или функций (к примеру, триггера отвечают только за свои области данных).

## **Резервное копирование базы данных**

Если над базой данных производят любое из ниже перечисленных структурных изменений, базы данных, непосредственно перед изменениями и после делается соответствующее копирование базы данных:

- Создание или удаление табличного пространства
- Добавление или переименование (перемещение) файла данных в существующем табличном пространстве
- Добавление, переименование(перемещение) или удаление группы или члена онлайн-журнала повторения.
- Если база данных работает в режиме ARCHIVELOG, то до и после структурного изменения базы данных требуется лишь резервное копирование

управляющего файла базы данных (с помощью команды ALTER DATABASE с опцией BACKUP CONTROLFILE). Можно скопировать и другие части базы данных.

- Если база данных работает в режиме NOARCHIVELOG, то непосредственно перед и после изменения базы данных требуется сделать полное копирование файла базы данных, включая все файлы данных, файлы журнала повторения и управляющие файлы.

Существует, по большому счету, два вида резервного копирования:

- Непротиворечивое (холодное) резервное копирование, ситуация, когда, копии создаются, в случае закрытой БД (close) для пользователей. Копия базы данных, созданной в автономном режиме, содержит: все файлы данных, журналы повторов и управляющие файлы. После останова БД, все файлы базы данной по средствам ОС копируются на один из backup дисков.
- Остановка экземпляра БД ✓ в режиме shutdown normal (игнорирование, новых подключений и ожидание отключение все зарегистрированных пользователей) или shutdown immediate (немедленное прерывание всех соединений, выполнение операции отката на всех транзакциях, ожидающих обработки)
- Копирование всех физических файлов, относящихся к базе данных, управляющие файлы, файлы журнала обновления и файлы базы данных.
- Закончить работу, перезагрузить базу данных
- Резервное(горячее) копирование в оперативном режиме, к примеру, когда БД работает в архивном режиме ARCHIVELOG, БД все время находиться в оперативном режиме таким образом доступна пользователям.
- Перевод табличного пространства в режим резервного копирования.
- Копирование всех файлов базы данных, связанных с табличным пространством.
- Выведение табличное пространство из режима резервного копирования.
- Повторение действий с первого по третье, пока не будет выполнено резервное копирование всех табличных пространств.
- Копирование управляющего файла.
- Копирование оперативного журнала обновления.

Сопоставление режима ARCHIVELOG и режима NOARCHIVELOG

В режиме ARCHIVELOG:

- Требуется дополнительное дисковое пространство

- Управление архивными журналами влечет за собой дополнительные административные непроизводительные затраты.
- Применимо горячее резервное копирование.
- В случае отказа носителя может быть выполнено полное восстановление базы данных.

В режиме NOARCHIVELOG:

- Не требуется дополнительное дисковое пространство или непроизводительные затраты.
- Может использоваться только холодное резервное копирование.
- В случае отказа теряется вся работа, выполненная со времени последнего резервного копирования.

Включение и выключение архивирования. Вы устанавливаете первоначальный режим архивирования базы данных во время ее создания. В большинстве случаев, во время создания базы данных вы можете выбрать режим по умолчанию NOARCHIVELOG, потому что нет необходимости архивировать информацию, генерируемую за этот период. После того как база данных создана, решите, нужно ли изменить первоначальный режим архивирования. После того, как база данных создана, всегда можно переключать режим архивирования базы данных. Однако, как правило, следует выбрать постоянный режим работы базы данных. Включение автоматического архивирования. Чтобы автоматическое архивирование заполненных групп было включено установите в TRUE значение параметра

LOG\_ARCHIVE\_START в файле параметров базы данных INIT.ORA:

LOG\_ARCHIVE\_START = TRUE это значение будет иметь эффект при очередном запуске базы данных.

Выключение автоматического архивирования. Вы можете выключить автоматическое архивирование журнала в любой момент. Однако, выключив автоматическое архивирование, вы должны вручную, периодически и своевременно, архивировать заполняемые группы журнала. Если база данных работает в режиме ARCHIVELOG, автоматическое архивирование выключено, а группы журнала заполняются, но не архивируются, то процесс LGWR не сможет повторно использовать неактивные группы журнала, чтобы продолжать запись информации повторения. Поэтому работа базы данных будет временно приостановлена до тех пор, пока не будет выполнено необходимое архивирование. Автоматическое архивирование может быть выключено как до, так и после запуска

инстанции. Если база данных работает в режиме ARCHIVELOG, то можно копировать индивидуальное табличное пространство или даже индивидуальный файл. Эта возможность полезна, если одна часть базы данных используется более интенсивно, чем другие, - например, табличное пространство SYSTEM или табличные пространства, содержащие сегменты отката. Если сбой диска повреждает один из таких файлов данных, для его реставрации может быть использована ранняя копия, и меньшее число изменений необходимо применить при прокрутке вперед, чтобы восстановить файл к состоянию на момент сбоя, т.е. время, затрачиваемое на восстановление, сокращается.

## **Используемая литература:**

[https://habr.com/hub/db\\_admins/](https://habr.com/hub/db_admins/)

[http://www.opennet.ru/docs/RUS/db\\_admin/#\\_Toc481223759](http://www.opennet.ru/docs/RUS/db_admin/#_Toc481223759)

<http://dit.isuct.ru/IVT/sitanov/Literatura/AdminInfSystem/Pages/Glava7.htm>

<http://5fan.ru/wievjob.php?id=36290>