

Лекція № 10
ТЕЗИ

Основи адміністрування СУБД MySQL

В кінці минулої лекції ми сформулювали список з 12-ти завдань адміністратора СУБД, які він повинен з періодичністю виконувати в процесі супроводу будь-якої СУБД. На сьогоднішній лекції розглянемо СУБД MySQL саме в контексті цих 12-ти завдань.

Особливості MySQL та її місце на ринку СУБД

Судя по всьому, именно простота работы с MySQL стала основной причиной широкой ее популярности среди пользователей.

LAMP і не тільки

16.01.2008 17:12 Sun Microsystems объявила о покупке компании MySQL AB
Компания Sun Microsystems объявила о достижении договоренности по покупке MySQL AB.

Сумма сделки составила примерно 800 миллионов долларов США за акции MySQL и 200 миллионов в качестве опциона.

MySQL будет полностью интегрирована в инфраструктуру разработки ПО, продаж и сервиса Sun. Руководитель MySQL AB, Marten Mickos, войдет в состав правления Sun Microsystems. Около 400 сотрудников MySQL постепенно вольются в единый коллектив Sun.

Установка програмного забезпечення СУБД MySQL

а також періодичне оновлення ПЗ сервера

дуже просто

На этапе инсталляции MySQL необходимо обязательно установить пароль для MySQL-пользователя root, поскольку сразу после установки права сервера не защищены. Предполагается, что каталог данных и база данных mysql с таблицей разрешений уже инициализированы. На компьютерах с UNIX для их инициализации достаточно запустить сценарий mysql_install_db. На компьютерах, работающих под управлением Windows, каталог данных и база данных mysql инициализируются посредством запуска программы Setup в дистрибутиве сервера. Итак, каталог и основная база данных проинициализированы, и сервер запущен.

Сразу после первой инсталляции MySQL на компьютере привилегии в таблице разрешений базы данных mysql устанавливаются следующим образом.

* Авторизоваться в качестве основного пользователя root с локального компьютера можно без пароля. Пользователь root обладает всеми возможными правами (включая административные), может выполнять любые операции. (Кстати, совпадение имен

суперпользователей MySQL и UNIX не является закономерностью Они никак друг на друга не влияют.)

- Права анонимного доступа предоставляются всем пользователям, подключающимся с локального компьютера к базе данных test или любой другой базе данных, имя которой начинается со слова test. Анонимные пользователи могут выполнять любые операции с такими таблицами, но не обладают привилегиями администратора.

В версиях MySQL 3.22 и выше установить пароль можно с помощью команды `mysqladmin`. Для этого достаточно ввести следующую команду, заменив ее часть "my password" реальным паролем:

```
mysqladmin -u root password "my password"
```

Во всех остальных версиях MySQL для этих целей можно воспользоваться программой `mysql` и непосредственно обновить таблицу разрешений `grant` в базе данных `mysql`:

```
mysql> -u root mysql
mysql> UPDATE user SET Password=PASSWORD("my password") WHERE User="root";
```

```
% mysqladmin -u root reload
```

Администратору следует помнить о двух целях, преследуемых при настройке процедуры запуска сервера MySQL.

* Желательно, чтобы сервер не обладал привилегиями пользователя `root`. В большинстве случаев рекомендуется ограничивать возможности всех процессов, не требующих прав доступа `root`. Не нужны эти права и демону `mysqld`.

Желательно, чтобы сервер все время работал под именем одного и того же пользователя. Весьма нелогично запускать сервер в разное время под именами разных пользователей, поскольку в этом случае файлы и каталоги с данными будут создаваться разными владельцами. В определенный момент времени это может привести к невозможности получения доступа к базам данных или таблицам. Постоянный запуск сервера от имени одного пользователя позволит избежать этой проблемы.

Стартові скріпти в freebsd linux

Методы запуска сервера

Сервер можно запускать вручную из командной строки или автоматически в процессе загрузки системы. Существуют три основных метода запуска сервера.

* Непосредственный вызов `mysqld`. Это, возможно, самый распространенный метод. Поэтому подробно рассматриваться далее он не будет. Достаточно лишь заметить, что в случае необходимости можно использовать команду `mysqld --help` для определения опций с другими методами запуска.

* Вызов сценария `safemysqld`. Этот сценарий в процессе своей работы пытается определить местоположение программы сервера и каталога данных, а затем запускает сервер с соответствующими опциями. Сценарий `safemysqld` записывает все сообщения об ошибках сервера в специальный файл ошибок, расположенный в каталоге данных. Кроме того, `safemysqld` следит за нормальной работой сервера и в случае сбоя перезагружает его. Этот сценарий наиболее часто применяется в BSD-версиях UNIX.

Вызов сценария `mysql.server`. Этот сценарий запускает сервер посредством запуска сценария `safemysqld`. Сценарий `mysql.server` предназначен для использования на компьютерах с системой запуска/завершения работы System V. Данная система включает несколько каталогов со сценариями, вызываемыми при входе или выходе с определенного уровня работы. С помощью соответствующих аргументов `start` и `stop` можно определить, что делать дальше: запустить сервер или остановить его работу.

Существует два способа определения дополнительных опций запуска, которые применяются при загрузке сервера. Во-первых, можно изменить используемый сценарий запуска (`safemysqld` или `mysql.server`) и **здать параметры непосредственно в строке вызова сервера**. Во-вторых, можно определить параметры собственно в конфигурационном файле. Профессионалы рекомендуют по возможности использовать для этих целей глобальные конфигурационные файлы. В системах UNIX и Windows этими файлами обычно являются `/etc/my.cnf`.

Завершение работы сервера

Для самостоятельного завершения работы сервера применяется команда `mysqladmin`:

```
% mysqladmin shutdown
```

Автоматическое завершение работы сервера также не требует выполнения каких-либо специальных действий. В UNIX BSD работа служб обычно завершается посредством отправки процессам сигнала TERM. Службы либо соответствующим образом на него отвечают, либо просто закрываются. Сервер `mysqld`, например, на получение такого сигнала реагирует закрытием. В системах System V, запуск сервера в которых производится с помощью сценария `mysql.server`, процедура завершения работы реализуется посредством вызова этого же сценария, но с аргументом `stop`. При этом, конечно же, предполагается, что сценарий `mysql.server` инсталлирован.

Способ хранения таблиц и баз данных

В MySQL таблице соответствует несколько файлов. Их имена совпадают с именем таблицы, а расширение определяет назначение файла. К примеру, файл с расширением `.frm` содержит описание структуры таблицы. Что касается баз данных, то они являются подкаталогами основного каталога данных (по умолчанию это `/usr/local/var`). Имя подкаталога соответствует имени базы данных. Это означает, что имена баз данных и таблиц отвечают тем же требованиям, которые предъявляются к именам файлов в данной системе. Скажем, файловая система `ext2` в Linux чувствительна к регистру символов, а `FAT32` в Windows — нет.

Начиная с версии 3.23.37 в MySQL поддерживаются семь типов таблиц. Три из них — Berkley DB, Gemini и Inno DB ориентированы на транзакции, а четыре — Heap, ISAM, Merge и MyIsam — нет. Транзакции являются относительно новым понятием в MySQL, но соответствующие функции для таблиц Berkeley DB и InnoDB существуют уже достаточно давно, что позволило включить их в стандартные бинарные дистрибутивы.

Стандартным типом таблиц в MySQL является тип MyISAM. Он возник на основе более старого типа ISAM, который все еще существует, хотя использовать его не рекомендуется. Переопределить установку по умолчанию позволяет опция TYPE инструкций CREATE TABLE и ALTER TABLE

Heap

MySQL хранит таблицы типа Heap в памяти, а не в файловой системе. Следовательно, доступ к ним осуществляется чрезвычайно быстро.

Доступ к резидентным таблицам имеют все пользователи. Эти таблицы уничтожаются при выключении сервера.

типы “движка” (engine) -

MyISAM

MyISAM – это стандартный тип таблиц в MySQL, если только в конфигурационном файле не задано иное. Для таблиц этого типа создан ряд специализированных утилит, позволяющих манипулировать табличными файлами. Сюда входят утилита `myisamchk` для проверки и восстановления таблиц и утилита `myisampack` для создания сжатых таблиц.

Таблицы MyISAM являются платформо-независимыми. Табличные файлы можно перемещать между компьютерами разных архитектур и разными операционными системами без всякого преобразования.

MySQL хранит счетчик подключений к таблице MyISAM. Когда таблица закрывается, счетчик сбрасывается в нуль. Если сервер неожиданно завершает работу, счетчик остается положительным числом. В таком случае в процессе перезапуска сервер обнаружит проблему. Это не означает, что таблица непременно повреждена, но подобная возможность существует.

InnoDB

СУБД InnoDB была разработана Хейкки Туури (Heikki Tuuri) из компании Innobase — финского производителя программного обеспечения, специализирующегося на технологии реляционных баз данных.

Ядро не существует само по себе, а является дополнением к MySQL.

На этапе компиляции MySQL поддержка InnoDB включается с помощью опции `--with-innodb`. Исходные коды InnoDB входят в исходный дистрибутив MySQL

В отличие от таблиц MyISAM, где для каждой таблицы создается один файл данных, данные InnoDB хранятся в больших совместно используемых файлах. Можно создать произвольное число файлов данных, но их нельзя будет удалить. Размер файлов определяется в конфигурационном файле. Если нужно уменьшить объем дискового пространства, занимаемого таблицами InnoDB, создайте резервные копии таблиц, после чего удалите все файлы InnoDB и позвольте программе MySQL восстановить их в соответствии с новыми установками конфигурационного файла. Журнальные файлы InnoDB можно безопасно удалить после остановки сервера. При повторном запуске сервера программа MySQL создаст журнальные файлы заново.

В листинге 6.1 приведен пример опций, которые необходимо добавить в конфигурационный файл в группу `[mysql]` чтобы активизировать таблицы InnoDB. Размер файлов здесь задан относительно небольшим, что вполне подходит для целей эксперимента. На практике используются файлы гораздо большего размера.

```
innodb_data_home_dir = /usr/local/var/innodb/  
innodb_data_file_path = ibdata1/ibdata1:100M; ibdata2/ibdata2:100M  
set-variable = innodb_mirrored_log_groups = 1  
innodb_log_group_home_dir = /disk2/innodb/log  
set-variable = innodb_log_files_in_group = 3  
set-variable = innodb_log_file_size = 16M  
set-variable = innodb_log_buffer_size = 8M  
innodb_flush_log_at_trx_commit = 1  
innodb_log_arch_dir = /disk2/innodb/log  
innodb_log_archive = 0  
set_variable = innodb_buffer_pool_size = 25M  
set_variable = innodb_additional_mem_pool_size = 5M  
set_variable = innodb_file_io_threads = 4  
set_variable = innodb_lock_wait_timeout = 50
```

Листинг 6.1. (html, txt)

После добавления опций в конфигурационный файл необходимо перезапустить сервер MySQL. Все необходимые файлы будут созданы автоматически. На это может уйти некоторое время, в зависимости от размера файлов и скорости жесткого диска.

Таблицы InnoDB блокируются на уровне записей. Это происходит без участия пользователей по мере выполнения инструкций в рамках транзакций.

На случай отмены транзакций ведется журнал транзакций. Он подвержен внутренней ротации, т.е. когда заполняются все записи, самые старые из них начинают удаляться.

Блокировки таблиц

В MySQL разрешается явно блокировать таблицы с помощью инструкции LOCK TABLES. Тем не менее, не рекомендуется делать это для таблиц тех типов, которые поддерживают транзакции. Блокировки и транзакции — это два разных способа решения проблемы одновременного доступа к таблице, поэтому нужно сделать выбор в пользу одного из них.

В зависимости от инструкции могут также применяться неявные блокировки. Например, инструкция UPDATE способна немедленно заблокировать таблицу, запретив доступ к ней другим потокам. Блокировки обоих типов защищены от возникновения тупиковых ситуаций, так что можно не волноваться по поводу отмены той или иной инструкции.

Можно заблокировать таблицу таким образом, чтобы разрешить другим потокам обращаться к ней для чтения. Это называется блокировкой чтения. Блокировка записи гарантирует текущему потоку монополярный доступ к таблице. Запросы на чтение откладываются до тех пор, пока не будут сняты все блокировки записи. Эту установку можно изменить с помощью флагов инструкций либо путем задания специальных серверных переменных.

Модель прав доступа до об'єктів в MySQL

(підтримка рахунків доступу користувачів до БД MySQL,
захист сервера від внутрішнього несанкціонованого доступу)

В обязанности администратора MySQL входит также создание и настройка учетных записей пользователей MySQL. В процессе этой настройки необходимо определить, какие пользователи будут иметь возможность подключения к серверу, откуда они смогут подключиться и что смогут делать после подключения. Два появившихся в MySQL 3.22.11 оператора упрощают эту задачу. Оператор GRANT создает пользователей MySQL и позволяет настроить их привилегии. Оператор REVOKE удаляет привилегии. Эти два оператора являются своего рода внешним интерфейсом для таблиц разрешений базы данных mysql и обеспечивают альтернативу непосредственному редактированию содержимого этих таблиц. Операторы grant и revoke работают с четырьмя следующими таблицами.

user Подключающиеся к серверу пользователи и все их глобальные привилегии

db Привилегии уровня базы данных

tables_priv Привилегии уровня таблицы

columns_priv Привилегии уровня столбца

Существует еще одна, пятая таблица разрешений (host), однако операторы grant и revoke не в состоянии ее обрабатывать.

Если оператор GRANT запускается для определенного пользователя, в таблице user для него создается новая запись. Если оператор определяет для пользователя какие-либо глобальные привилегии (привилегии администратора или привилегии, применяемые сразу ко всем базам данных), они также записываются в таблицу user. Права обработки базы данных, таблицы или столбца записываются соответственно в таблицы db, tables_priv и column_priv.

Оператор GRANT имеет следующий синтаксис:

```
GRANT privileges (columns)
ON what
TO user IDENTIFIED BY "password"
WITH GRANT OPTION
```

Для успешного его выполнения обязательно нужно правильно определить следующую информацию:

* privileges (привилегии). Привилегии присваиваются определенному пользователю. Используемые в операторе grant спецификаторы привилегий описаны в представленной ниже таблице.

Спецификатор привилегий	Разрешенная операция
user	Подключающиеся к серверу пользователи и все их глобальные привилегии
alter	Изменение таблиц и индексов
create	Создание баз данных и таблиц
delete	Удаление существующих записей из таблиц
drop	Удаление баз данных и таблиц
index	Создание и удаление индексов
insert	Вставка новых записей в таблицы
references	Не используется
select	Извлечение существующих записей из таблиц
update	Изменение существующих записей таблиц
file	Чтение и запись файлов сервера
process	Просмотр информации о внутренних потоках сервера и их удаление

reload Перезагрузка таблиц разрешений или обновление журналов, кэша компьютера или кэша таблицы
shutdown Завершение работы сервера
all Все операции. Аналог — all privileges
usage Полное отсутствие привилегий

Резервне копіювання БД

Администратор должен обязательно резервировать базы данных на случай повреждения или потери данных. Только благодаря резервированию все таблицы могут быть восстановлены до прежнего состояния в случае сбоя в работе системы. Кроме того, не исключен вариант, когда резервирование может оказаться единственным путем отступления, если какой-либо неопытный пользователь случайно выполнит операторы DROP database или drop table. Иногда сбой может произойти по вине собственно администратора MySQL.

Существует два основных способа резервирования баз данных: использование программы mysqldump и непосредственное копирование файлов базы данных (с помощью команд cp, tar или cpio). Каждый метод имеет свои преимущества и недостатки.

*

Программа mysqldump тесно взаимодействует с сервером MySQL. Методы непосредственного копирования являются внешними по отношению к серверу и требуют проверки, чтобы клиенты не пытались изменить таблицы баз данных в процессе копирования. Эта же проблема возникает при использовании для резервирования баз данных средств резервирования файловой системы. Если в процессе резервирования кто-то из пользователей изменяет таблицы, их файлы окажутся несовместимыми и не подлежащими восстановлению. Разница между резервированием файловой системы и непосредственным копированием файлов заключается в том, что в первом случае имеется возможность управлять расписанием резервирования.

*

Программа mysqldump медленней резервирует данные, чем методы непосредственного копирования.

*

Программа mysqldump создает простые текстовые файлы, которые можно легко переносить на другие компьютеры, даже с другой аппаратной архитектурой. Копируемые вручную файлы не могут переноситься на другие компьютеры, если, конечно, не используется специальный формат хранения MyISAM. ISAM-таблицы могут копироваться только между компьютерами с подобной архитектурой.

При резервировании базы данных с помощью программы mysqldump создается файл, который состоит из создающих таблицы операторов create table и операторов insert с данными строк таблиц. Другими словами, в процессе работы программа mysqldump создает набор операторов, которые впоследствии можно отправить серверу mysql для воссоздания базы данных.

Для резервирования целой базы данных в одном текстовом файле достаточно запустить следующую команду:

```
% mysqldump samp_db > /usr/archives/mysql/samp_db.1999-10-02
```

В результате ее выполнения будет создан файл с командами sql.

Опції:

-add-drop-table

Комбинация опций -flush-logs и —lock-tables полезна для контроля состояния базы данных. Первая опция закрывает и повторно открывает файлы журналов обновлений, а вторая —блокирует все резервируемые таблицы.

#

Команда mysqldump по умолчанию перед записью таблицы в архив считывает ее всю в память. В этом, однако, нет необходимости. Более того, подобная обработка больших таблиц вообще может привести к сбою. Поэтому администратор может воспользоваться опцией —quick, определяющей построчное считывание и запись информации. Чтобы еще больше оптимизировать процесс резервирования, вместо опции —quick можно применить опцию —opt. Она, в свою очередь, активизирует все остальные опции, ускоряющие считывание и резервирование данных.

Выполнение резервирования с помощью опции —opt — наиболее распространенный (благодаря скорости) метод выполнения. Однако следует проявлять осторожность, поскольку опция —opt оптимизирует процедуру резервирования, закрывая на время доступ к базе данных. Она блокирует все таблицы сразу, запрещая внесение каких-либо изменений. Эффект от применения этой опции заметить очень легко. Попробуйте запустить команду резервирования с этой опцией днем, во время наиболее частого использования базы данных. Пользователи не заставят себя долго ждать, и скоро телефон администратора начнет звонить не переставая.

#

Эффекта, прямо противоположного результатам опции —opt, можно достичь с помощью опции —delayed. Эта опция заставляет команду mysqldump записывать в файл архива операторы INSERT DELAYED вместо операторов INSERT. Опция —delayed оказывается весьма полезной, если при загрузке файла архива в другую базу данных администратор желает уменьшить влияние этой операции на выполнение текущих запросов.

Моніторинг стану сервера

(підтримка/перегляд лог-файлів чи іншої інформації про роботу сервера БД)

Зміна параметрів роботи сервера, оптимізація використання ресурсів

швидкодія, продуктивність, локалізація і т.п.

Реплікація в MySQL

буквально кілька слів про реплікацію.

Термин "репликация" может означать как простое копирование базы данных на другой компьютер, так и интерактивное обновление подобной второй базы данных при внесении изменений в основную базу данных. Если необходимо просто скопировать базу данных на другой компьютер, можно воспользоваться одним из описанных выше

методов. Первые признаки возможностей интерактивного обновления появились только в версии MySQL 3.23. Пока они находятся на стадии разработки

Захист СУБД MySQL на рівні файлової системи і мережевих засобів

Відновлення баз даних MySQL після збоїв

Типові операції по превентивній підтримці сервера MySQL