

Лекція № **11**
ТЕЗИ

Основи адміністрування СУБД MySQL
(Продовження)

Модель прав доступу до об'єктів в 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.dump
```

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)

Логи (детальність залежить від параметрів запуску сервера)

mysqladmin — утиліта для моніторингу стану сервера, для управління станом сервера.

```
mysqladmin proc -  
mysqladmin stat  
mysqladmin variables  
mysqladmin reload  
mysqladmin create db_name  
mysqladmin drop db_name
```

Утиліта для виконання адміністративних операцій. Її синтаксис:

```
shell> mysqladmin [ПАРАМЕТРИ] command [command-option] command ...
```

Список опцій, підтримуємих вашою конкретною версією `mysqladmin`, можна отримати, виконавши команду `mysqladmin -help`.

Текущая версия `mysqladmin` підтримує наступні команди:

- * `create databasename`
Создать новую базу данных.
- * `drop databasename`
Удалить базу данных и все ее таблицы.
- * `extended-status`
Выдает расширенный отчет о состоянии сервера (более полный, чем при команде `status`).
- * `flush-hosts`
Сбросить и перезагрузить хосты.
- * `flush-logs`
Сбросить на диск и переоткрыть все журналы.
- * `flush-tables`
Закрыть все открытые таблицы.
- * `flush-privileges`
Перечитать таблицы привилегий.
- * `kill id,id,...`
Завершить потоки `mysql` с указанными `thread-id`.
- * `password`
Установить новый пароль для сервера баз данных. Изменить старый пароль на **НОВЫЙ**.
- * `ping`
Проверить, работает ли сервер `mysqld`.
- * `processlist`
Показать список активных потоков на сервере.

* reload

Перезагрузить таблицы привилегий.

* refresh

Выполнить все табличные операции, находящиеся в буфере, закрыть и открыть заново все системные журналы.

* shutdown

Завершить работу сервера баз данных.

* slave-start

Запустить подчиненный дублирующий поток.

* slave-stop

Остановить подчиненный дублирующий поток.

* status

Выдает краткий отчет о состоянии сервера.

* variables

Вывести доступные для использования переменные.

* version

Вывести данные о версии сервера.

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

Повреждения в таблицах MyISAM происходят вследствие событий, которые невозможно избежать. Различные аппаратные сбои могут оказать самое непредсказуемое влияние на базу данных. Например, если жесткий диск выйдет из строя, данные окажутся полностью потерянными. Неожиданное выключение системы из-за сбоя питания может привести к тому, что изменения в таблицу будут внесены не полностью. Даже если уничтожить серверный процесс по команде kill, у него не будет возможности корректно завершить свою работу.

Если найдена поврежденная таблица, потратьте время на выяснение причин, вызвавших повреждение. Вообще говоря, в MySQL таблицы редко оказываются поврежденными.

Существуют два способа проверки и восстановления таблиц. Первый — с помощью специальных инструкций, второй — с помощью утилиты `myisamchk`. Соответствующие инструкции называются `CHECK TABLE`, `REPAIR TABLE` и `OPTIMIZE TABLE`. Они достаточно удобны, поскольку выполняются в рамках серверного процесса. В этом смысле они ничем не отличаются, к примеру, от инструкции `SELECT`. Утилита `myisamchk` обладает рядом дополнительных возможностей, которые в ряде ситуаций оказываются весьма удобными.

Необходимость проверки таблицы может быть вызвана тем, что утилиты, обращающиеся к таблице, начинают себя странно вести. Например, вводимые запросы не завершаются или выдаются неожиданные сообщения об ошибках. Если при обращении к таблице возвращается номер ошибки, воспользуйтесь утилитой `pergror`, которая отображает поясняющее сообщение, соответствующее данному номеру.

Частота проверок базы данных зависит от степени доверия к серверу. Разработчики MySQL рекомендуют делать это хотя бы раз в неделю, но если есть возможность выполнять процедуру проверки каждую ночь, то шансы на заблаговременное обнаружение ошибки возрастают. С помощью демона `cron` или программы-планировщика можно составить график проверок таким образом, чтобы они запускались в часы наименьшей активности системы. Сохраняйте результаты проверок в журнальном файле или направляйте их самому себе по электронной почте.

Использование `mysqlcheck` для сопровождения и аварийного восстановления таблиц.

Начиная с версии MySQL 3.23.38, можно применять новый инструмент для проверки и восстановления MyISAM-таблиц. Отличие `mysqlcheck` от `myisamchk` состоит в том, что утилита `mysqlcheck` должна использоваться при работающем сервере `mysqld`, в то время как `myisamchk` — при остановленном. Преимущество же заключается в том, что теперь не нужно останавливать сервер для проверки или восстановления таблиц.

Утилита `mysqlcheck` использует соответствующие команды MySQL-сервера `CHECK`, `REPAIR`, `ANALYZE` и `OPTIMIZE` удобным для пользователя образом.

Существует три альтернативных способа запуска `mysqlcheck`:

```
shell> mysqlcheck [OPTIONS] database [tables]
shell> mysqlcheck [OPTIONS] -databases DB1 [DB2 DB3...]
shell> mysqlcheck [OPTIONS] -all-databases
```

Перелік основних утиліт і сценаріїв сервера MySQL

`myisamchk`

Утилита, применяемая для описания, проверки, оптимизации и восстановления таблиц MySQL. Поскольку утилита `myisamchk` имеет много функций, она описывается отдельно.

`mysqld`

Сам сервер MySQL, демон. Он должен работать постоянно.

mysql_install_db

Создает таблицы привилегий MySQL с заданными по умолчанию привилегиями. Это обычно делается только один раз, при первой инсталляции MySQL в системе.

safe_mysqld

оболочка mysqld

это сценарий, с помощью которого рекомендуется запускать демон mysqld на Unix. safe_mysqld служит для реализации неких дополнительных функций безопасности для mysqld, таких, как возможность перезапуска сервера при обнаружении ошибки и записи в журнал информации о процессе работы mysqld.

mysqld_multi

предназначена для управления несколькими процессами mysqld, работающих на различных сокетах Unix и портах TCP/IP.

myisampack

используется для сжатия таблиц MyISAM (*только для чтения*) сжимает каждый столбец в таблице по отдельности. Информация, необходимая для декомпрессии столбцов, считывается в память при открытии таблицы. В результате обеспечивается более высокая производительность при доступе к отдельным записям, поскольку нужно распаковывать только одну запись, а не значительно больший по размеру дисковый блок. В среднем myisampack сжимает файл данных на 40%-70%.

* После сжатия таблица доступна в режиме только для чтения. Это удобно, скажем, для записи на CD.

* myisampack может также сжимать столбцы с типами BLOB или TEXT.

Утилиту myisampack можно запустить следующим образом:

```
shell> myisampack [options] filename ...
```

Каждое имя файла (filename) должно быть именем индексного файла (.MYI). Если вы не находитесь в директории базы данных, необходимо указать полный путь к файлу. Допускается опускать расширение .MYI.

В приведенном ниже списке дано краткое описание основных (е й інші) **клиентских программ** MySQL.

mysqlaccess

Сценарий, который проверяет привилегии доступа для комбинации значений хоста, пользователя и базы данных

mysqladmin

Утилита для выполнения административных функций, таких, как создание или удаление баз данных, перезагрузка таблиц привилегий, запись на диск содержимого таблиц, находящегося в буфере, повторное открытие файлов журналов. Утилита mysqladmin может также использоваться для получения информации с сервера о номере версии, процессах и состоянии сервера

mysqldump

Выводит содержимое базы данных MySQL в виде файла с SQL-операторами или в виде текстовых файлов с символом табуляции в качестве разделителя

mysqlimport

Импортирует текстовые файлы в соответствующие таблицы, используя команду
LOAD DATA INFILE

mysqlshow

Отображает информацию о существующих базах данных, таблицах, полях и
индексах