## **Лабораторная работа № 8 Управление правами пользователей**

**Теоретические сведения**

СУБД MySQL является многопользовательской средой, поэтому для доступа к таблицам БД могут быть созданы различные учетные записи с разным уровнем привилегий. Учетной записи редактора можно предоставить привилегии на просмотр таблицы, добавление новых записей и обновление уже существующих. Администратору БД можно предоставить более широкие полномочия (возможность создания таблиц, редактирования и удаления уже существующих). Для пользователя БД достаточно лишь просмотра таблиц.

Рассмотрим следующие вопросы:

* создание, редактирование и удаление учетных записей пользователей;
* назначение и отмена привилегий.

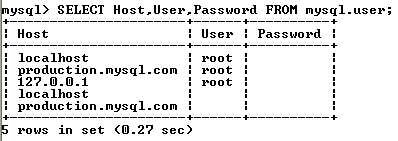
Учетная запись является составной и принимает форму *'username' @ 'host'*, где *username* – имя пользователя, a *host* – наименование хоста, с которого пользователь может обращаться к серверу. Например, записи *'root' @ '127.0.0.1'* и *'wet' @ '62.78.56.34'* означают, что пользователь с именем *root* может обращаться с хоста, на котором расположен сервер, a *wet* – только с хоста с IP-адресом 62.78.56.34.

IP-адрес 127.0.0.1 всегда относится к локальному хосту. Если сервер и клиент установлены на одном хосте, то сервер слушает соединения по этому адресу, а клиент отправляет на него SQL-запросы.

IP-адрес 127.0.0.1 имеет псевдоним *localhost*, поэтому учетные записи вида *'root' @ '127.0.0.1'* можно записывать в виде *'root' @ 'localhost'*.

Число адресов, с которых необходимо обеспечить доступ пользователю, может быть значительным. Для задания диапазона в имени хоста используется специальный символ "%". Так, учетная запись *'wet' @ '%'* позволяет пользователю *wet* обращаться к серверу MySQL с любых компьютеров сети.

Все учетные записи хранятся в таблице *user* системной базы данных с именем *mysql*. После первой инсталляции содержимое таблицы *user* выглядит так, как показано в листинге.



**Создание новой учетной записи**. Создать учетную запись позволяет оператор

*CREATE USER 'username' @ 'host' [IDENTIFIED BY [PASSWORD] 'пароль'];*

Оператор создает новую учетную запись с необязательным паролем. Если пароль не указан, в его качестве выступает пустая строка. Разумно хранить пароль в виде хэш-кода, полученного в результате необратимого шифрования. Чтобы воспользоваться этим механизмом авторизации, необходимо поместить между ключевым словом *identified by* и паролем ключевое слово *password*.

**Удаление учетной записи**. Удалить учетную запись позволяет оператор

*DROP USER 'username' @ 'host';*

**Изменение имени пользователя в учетной записи**. Осуществляется с помощью оператора

*RENAME USER старое\_имя* *TO новое\_имя;*

**Назначение привилегий**. Рассмотренные выше операторы позволяют создавать, удалять и редактировать учетные записи, но они не позволяют изменять привилегии пользователя – сооб­щать MySQL, какой пользователь имеет право только на чтение информации, какой на чтение и редактирование, а кому предоставлены права изменять структуру БД и создавать учетные записи.

Для решения этих задач предназначены операторы *grant* (назначает привилегии) и *revoke* (удаляет привилегии). Если учетной записи, котораяпоказана в операторе *grant*, не существует, то она автоматически создается. Удаление всех привилегий с помощью оператора *revoke* не приводит к авто­матическому уничтожению учетной записи. Для удаления пользователя необходимо воспользоваться оператором *drop user*.

В простейшем случае оператор *grant* выглядит следующим образом:



Данный запрос создает пользователя с именем *wet* и паролем *pass*, который может обращаться к серверу с локального хоста (*localhost*) и имеет все права (*all*) для всех баз данных (\*.\*). Если такой пользователь существует, то его привилегии будут изменены на *all*.

Вместо ключевого слова *all* можно использовать любое из ключевых слов, представленных в табл. 9. Ключевое слово *on* в операторе *grant* задает уровень привилегий, которые могут быть заданы на одном из четырех уровней, представленных в табл. 10. Для таблиц можно установить только следующие типы привилегий: *SELEСТ, INSERT, UPDATE, DELETE, CREATE, DROP, GRANT OPTION, INDEX* и *ALTER*. Это следует учитывать при использовании конструкции *grant all*, которая назначает привилегии на текущем уровне. Так, запрос уровня базы данных *grant all on db.\** не предоставляет никаких глобальных привилегий.

**Отмена привилегий**. Для отмены привилегий используется оператор *revoke*:



Оператор *revoke* отменяет привилегии, но не удаляет учетные записи, для их удаления необходимо воспользоваться оператором *drop USER*.

Таблица 9

|  |  |
| --- | --- |
| **Привилегия** | **Операция, разрешенная привилегией** |
| *ALL*  *[PRIVILEGES]* | Комбинация всех привилегий, за исключением привилегии *GRANT* *option*, которая задается отдельно |
| *ALTER* | Позволяет редактировать таблицу с помощью оператора *ALTER TABLE* |
| *ALTER ROUTINE* | Позволяет редактировать или удалять хранимую процедуру |
| *CREATE* | Позволяет создавать таблицу при помощи оператора *CREATE TABLE* |
| *CREATE ROUTINE* | Позволяет создавать хранимую процедуру |
| *CREATE TEMPORARY TABLES* | Позволяет создавать временные таблицы |
| *CREATE USER* | Позволяет работать с учетными записями c помощью *CREATE USER, DROP USER, RENAME USER* и *REVOKE ALL PRIVILEGES* |
| *CREATE VIEW* | Позволяет создавать представление с помощью оператора *CREATE VIEW* |
| *DELETE* | Позволяет удалять записи при помощи оператора *delete* |
| *DROP* | Позволяет удалять таблицы при помощи оператора *DROP TABLE* |
| *EXECUTE* | Позволяет выполнять хранимые процедуры |
| *INDEX* | Позволяет работать с индексами, в частности, использовать операторы  *CREATE INDEX* и *DROP INDEX* |
| *INSERT* | Позволяет добавлять в таблицу новые записи оператором *insert* |
| *LOCK TABLES* | Позволяет осуществлять блокировки таблиц при помощи операторов *LOCK TABLES* и *UNLOCK TABLES*. Для вступления в действие этой привилегии должна быть установлена привилегия *SELECT* |
| *select* | Позволяет осуществлять выборки таблиц оператором *SELECT* |
| *Show databases* | Позволяет просматривать список всех таблиц на сервере при помощи оператора *show databases* |
| *Show view* | Позволяет использовать оператор *show create view* |
| *UPDATE* | Позволяет обновлять содержимое таблиц оператором *UPDATE* |
| *USAGE* | Синоним для статуса «отсутствуют привилегии» |
| *GRANT OPTION* | Позволяет управлять привилегиями других пользователей, без данной привилегии невозможно выполнить операторы *grant* и *REVOKE* |

Таблица 10

|  |  |
| --- | --- |
| **Ключевое**  **слово *ON*** | **Уровень** |
| *ON \*.\** | Глобальный уровень – пользователь с полномочиями на глобальном уровне может обращаться ко всем БД и таблицам, входящим в их состав |
| *ON db.\** | Уровень базы данных – привилегии распространяются на таблицы базы данных *db* |
| *ON db.tbl* | Уровень таблицы – привилегии рас­пространяются на таблицу *tbl* базы данных *db* |
| *ON db.tbl* | Уровень столбца – привилегии касаются отдельных столбцов в таблице *tbl* базы данных *db*. Список столбцов указывается в скобках через запятую после ключевых слов *select, insert, update* |

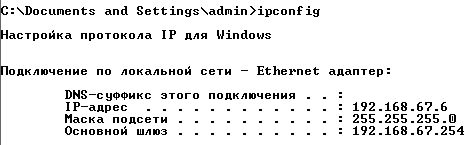
**Практическая работа**

При выполнении лабораторной работы необходимо:

* создать учетную запись нового пользователя и наделить его определенными привилегиями; пользователи и права доступа были заданы в отчете №1
* составить отчет по лабораторной работе.

**Пример выполнения работы**

Для работы выберем два компьютера, подключенных к локальной сети. На одном необходимо развернуть сервер MySQL, на другой – скопировать клиент командной строки *mysql.exe*. Определим IP-адрес сервера:



Создадим новую учетную запись, позволив пользователю *user1* обращаться к серверу MySQL с любых компьютеров сети:



Назначим этому пользователю привилегии глобального уровня:



На клиентском компьютере в командной строке (например, с помощью FAR), запустим клиент командной строки в следующем формате:



Наблюдаем отклик удаленного сервера и работаем с ним как обычно:

