***Лабораторная работа №6***

**Тема:** Знакомство с MySQL. Организация доступа к БД.

**Цель:** Получить практические навыки создания пользователей и настройки им привилегий.

**Задание:**

1. Изучить теоретическую часть.
2. Создать МИНИМУМ 4 пользователей с разными уровнями доступа к базе данных.
3. Оформить отчет.

***Теоретическая часть***

**Привилегии в MySQL**

При подключении к серверу MySQL пользователь указывает имя и пароль, а сервер проверяет, имеет ли клиент право получить доступ к серверу, и в случае успеха наделяет пользователя привилегиями, исходя из данных о привилегиях конкретного пользователя. Эти данные хранятся в системной базе mysql.

Проверка пользователя осуществляется с помощью трех полей таблицы user (host, user и password) базы mysql. Сервер устанавливает соединение только в том случае, если находит в таблице user запись, в которой имя пользователя и пароль совпадают с введенными значениями. Поле password может быть пустым. Это не означает, что в данном случае подходит любой пароль. Если поле пароля пусто, пользователь должен быть подсоединен без указания какого-либо пароля.

Непустые значения в поле password представляют собой зашифрованные пароли. В MySQL пароли не хранятся в виде открытого текста, который может прочитать кто угодно, пароль, который вводится пользователем при попытке подсоединения, шифруется (с помощью функции password()). В дальнейшем зашифрованный пароль используется клиентом и сервером в процессе проверки его правильности.

При переходе к версии MySQL 4.1.3 произошло очень важное изменение. Дело в том, что зашифрованный пароль может быть разной длины. Длина зашифрованного пароля зависит от алгоритма шифрования, и чем больше длина, тем труднее отгадать пароль. Шифрацию пароля осуществляет сам сервер при создании новой учетной записи. Пользователь вводит пароль при соединении с сервером, клиентская программа его шифрует и отправляет на сервер в зашифрованном виде. Сервер не расшифровывает пароль, а сравнивает присланную ему зашифрованную строку с хранящейся у него строкой пароля. Алгоритм шифрования таков, что шифрация одинаковых слов всегда даст один и тот же результат.

В версии MySQL 4.0 длина шифра составляла 16 символов, шифровали пароли клиентская утилита mysql и другие клиентские утилиты, также шифровали пароли и функции РНР для доступа к MySQL.

Начиная с MySQL 4.1.3, изменен алгоритм шифрации – длина шифра теперь составляет 41 символ. Вы можете убедиться в этом, дав команды:

mysql>USE mysql

mysql>SELECT password FROM user;

Полезно посмотреть, что содержится в этой таблице. Поупражняйтесь в этом, давая команды select для выборки различных полей. Сначала дайте команду:

mysql>SHOW TABLES;

Потом постарайтесь разобраться, какую информацию содержат таблицы базы mysql.

Что из всего этого следует? Первое: старые клиентские утилиты не годятся для работы с новым сервером MySQL 5. Поэтому работайте только с клиентскими утилитами из того пакета, из которого вы ставили сервер.

И второе: для работы с сервером MySQL 4.1 и более новыми версиями, в частности, 5.0, требуется новая библиотека функций РНР, это расширение mysqli – новое семейство функций для работы с сервером MySQL.

После установления соединения сервер приступает к выполнению второго этапа. Для каждого поступающего запроса сервер проверяет, имеется ли у вас достаточно привилегий для выполнения запроса. Информация о привилегиях находится в таблицах привилегий – user, db, host, tables\_priv или columns\_priv.

Список привилегий представлен в табл. 4. Сервер MySQL считывает содержимое этих таблиц во время запуска и в случаях, когда изменения в привилегиях вступают в силу.

Таблица 4. Привилегии MySQL

|  |  |
| --- | --- |
| **Привилегия** | **К чему применяется** |
| ALTER | Таблицы |
| DELETE | Таблицы |
| INDEX | Таблицы |
| INSERT | Таблицы |
| SELECT | Таблицы |
| UPDATE | Таблицы |
| CREATE | Базы данных, таблицы или индексы |
| DROP | Базы данных или таблицы |
| GRANT | Базы данных или таблицы |
| SHOW DATABASES | Администрирование сервера |
| SHUTDOWN | Администрирование сервера |

Привилегии SELECT, INSERT, UPDATE И DELETE позволяют выполнять операции над строками таблиц баз данных. Привилегия index обеспечивает создание или уничтожение индексов. Привилегия alter позволяет использовать команду alter table. Привилегии create и drop позволяют создавать новые или уничтожать существующие базы данных и таблицы. Привилегия grant позволяет вам предоставлять другим пользователям привилегии, которыми обладаете вы сами.

Остальные привилегии используются для операций администрирования.

Команда shutdown завершает работу сервера.

При запуске сервера MySQL все таблицы назначения привилегий загружаются в память, и с этого момента привилегии вступают в силу.

Команды grant и revoke позволяют системным администраторам создавать пользователей MySQL, а также предоставлять права пользователям или лишать их прав на четырех уровнях.

1. Глобальный уровень.

Глобальные привилегии применяются ко всем базам данных на указанном сервере. Эти привилегии хранятся в таблице mysql.user.

2. Уровень базы данных.

Привилегии базы данных применяются ко всем таблицам указанной базы данных. Эти привилегии хранятся в таблицах mysql .db и mysql .host.

3. Уровень таблицы.

Привилегии таблицы применяются ко всем столбцам указанной таблицы. Эти привилегии хранятся в таблице mysql. tables\_priv.

4. Уровень столбца.

Привилегии столбца применяются к отдельным столбцам указанной таблицы. Эти привилегии хранятся в таблице mysql. columns\_priv.

Команда grant позволяет добавлять новых пользователей сервера:

mysql> GRANT SELECT, INSERT ON taxi. \* TO Mike@localhost

-> IDENTIFIED BY 'secret';

Эта команда дает права insert и select на базу taxi (на все ее таблицы – это задается символом звездочки) пользователю Mike, который подключается локально (localhost), причем при подключении к серверу пользователь должен указать пароль 'secret’. Дайте такую команду, а потом посмотрите, что изменилось в таблице user базы mysql.

Вот еще пример использования команды grant:

mysql> GRANT ALL PRIVILEGES ON \*.\* TO Anna@"%"

-> IDENTIFIED BY 'Some\_pass' WITH GRANT OPTION;

Здесь Anna получает все права (all privileges) на все таблицы всех баз сервера (\*. \*), причем подключаться она может как локально, так и удаленно – через сеть (это указано символом % после Anna@). Вдобавок Anna может передавать свои права другим пользователям (with grant option), т. е. сама может давать команду grant и создавать новых пользователей системы.

mysql> GRANT USAGE ON \*.\* TO dummy@localhost;

Пользователь dummy, созданный в этом примере, может подсоединяться к серверу без пароля, но только с локального компьютера. Привилегия usage означает, что пользователя в системе завели, но прав у него никаких нет. Такое может потребоваться, когда человек принят на работу, но обязанности его еще не определены. Предполагается, что относящиеся к базам данных привилегии будут назначены позже.

Команда set может устанавливать различные опции, влияющие на работу сервера или клиента. Пароль для пользователя root также можно задать с помощью этой команды:

mysql> SET PASSWORD FOR root @ localhost = PASSWORD ('new\_password1) ;

Пользователи могут работать и непосредственно с таблицами назначения привилегий:

mysql> use mysql;

mysql> UPDATE user SET Password=PASSWORD)'new\_password1)

-> WHERE user='root';

После этого следует дать команду, заставляющую сервер перечитать таблицы привилегий – тогда они вступят в силу:

mysql> FLUSH PRIVILEGES;

Если пароль задается при помощи оператора grant ... identified by, нет необходимости использовать функцию password(). Эта команда самостоятельно производит шифрацию пароля, поэтому пароль следует указывать в простом текстовом виде, например, таким образом:

mysql> GRANT USAGE ON \*.\* TO Jeffreys@"%" IDENTIFIED BY 'biscuit';

Изменения, которые вносятся в таблицы назначения привилегий при помощи команд grant, revoke или set password, учитываются сервером немедленно.

При создании нового пользователя в MySQL 5 не удастся создать пользователя с пустым паролем – система не позволит из соображений безопасности.

**Содержание отчета:**

* Тема;
* Цель;
* Описание каждого пользователя с перечислением его привилегий, скрипт запроса на языке SQL, результат работы запроса.