      MySQL授权系统主要通过五个表（user、db、host、tables\_priv和columns\_priv）来实现，其中用于访问数据库的各种用户信息都保存在mysql库的user表中。账户权限信息被存储在mysql数据库的user、db、host、tables\_priv、columns\_priv和procs\_priv表中。

      user表——该表决定是否允许用户连接到服务器。如果允许连接，权限字段则为该用户的全局权限。

      db表——用于决定哪些用户可以从哪些主机访问哪些数据库。包含在db表中的权限适用于这个表标识的数据库。

      host表——当您想在db表的范围之内扩展一个条目时，就会用到这个表。举例来说，如果某个db允许通过多个主机访问的话，那么超级用户就可以让db表内将host列为空，然后用必要的主机名填充host表。

      tables\_priv表——该表与db表相似，不同之处是它用于表而不是数据库。这个表还包含一个其他字段类型，包括timestamp和grantor两个字段，用于存储时间戳和授权方。在本文后面我们会对这个表做进一步的讲解。

      columns\_priv——该表作用几乎与db和tables\_priv表一样，不同之处是它提供的是针对某些表的特定列的权限。

权限表的存取过程是：

    1)  先从user表中的host、 user、 password这3个字段中判断连接的IP、用户名、密码是否存在表中，存在则通过身份验证；

2)   通过权限验证，进行权限分配时，按照user、db、tables\_priv、columns\_priv的顺序进行分配。即先检查全局权限表 user，如果user中对应的权限为Y，则此用户对所有数据库的权限都为Y，将不再检查db, tables\_priv,columns\_priv；如果为N，则到db表中检查此用户对应的具体数据库，并得到db中为Y的权限；如果db中为N，则检 查tables\_priv中此数据库对应的具体表，取得表中的权限Y，以此类推。

1．创建用户

1)        可以使用CREATE USER用于创建新的MySQL账户, CREATE USER会在没有权限的mysql.user表中创建一个新记录。如果 账户已经存在，则出现错误。

CREATE USER *user* [IDENTIFIED BY [PASSWORD] '*password*'] [, *user* [IDENTIFIED BY [PASSWORD] '*password*']] ...

        mysql>create user test1@'localhost' identified by '12345';    --test1只能本地访问  
        mysql>create user test2@'%' identified by '12345';              --test2可以远程访问

        mysql>create user [test3@’172.21.78.89](mailto:test3@'172.21.78.89)’ identified by ‘12345’;  --test3仅从172.21.78.89上访问

                     create user [test3@172.21.78.89](mailto:test3@172.21.78.89) identified by ‘12345’;

        mysql>create user [test4@172.21.78.%](mailto:test4@172.21.78.%25) identified by ‘12345’;   --test4可以从172.21.78.\*网段上访问

2)        也可以使用grant语句，GRANT语句的主要用途是来给帐户授权的，但也可用来建立新帐户并同时授权。

        mysql>grant select,insert,update,delete on mydb.\* to test1@localhost identified by "123456"；

        可以在hostname中指定通配符。例如user\_name@'%.loc.gov'适用于在loc.gov域中的任何主机的user\_name。同时user\_name@'144.155.166.%'适用于144.155.166 C级子网中的任何主机的user\_name。简单形式user\_name是user\_name@'%'的同义词。

3)        可以直接用INSERT语句创建相同的账户，然后使用FLUSH PRIVILEGES告诉服务器重载授权表。

        mysql> insert into mysql.user(Host,User,Password) values("localhost","test3",password("12345"));

        mysql>flush privileges;

第一种和第三种创建用户后还需要进行授权。

2．数据库用户授权

        GRANT权限列表 ON 库名.表名TO 用户名@来源地址[ IDENTIFIEDBY ‘密码‘]

使用GRANT语句的注意事项：

■ 权限列表：用于列出授权使用的各种数据库操作，以逗号分隔。All表示所有权限；

■ 库名.表名：用于指定授权操作的库和表的名称，其中可以使用“\*”。如：使用“mydb.\*”表示授权操作的对象为mydb库中的所有表。

■ 用户名@来源地址：用于指定用户名称和允许访问的客户机地址，即谁能连接、能从哪里连接。来源的地址可以是域名、IP地址，还可以使用“%”通配符，表示某个区域或网段内的所有地址。如：“%.example.com”、“172.168.10.%”等

■ IDENTIFIED BY：设置用户连接数据库时所使用的密码字符串。在新建用户时，若省略则用户的密码将为空。

■ 使用GRANT语句授权的用户记录，会保存到mysql库的user、db、host、tables\_priv等。

例：添加一个名为test1的数据库用户，并允许其从本机访问，对mydb库中的所有表具有查询权限，验证密码为“12345”。

       mysql> GRANT select ON mydb.\* TO 'test1'@'localhost' IDENTIFIED BY '12345';

例：增加一个test2用户，密码为12345，可以在任何主机上登录，并对所有数据库有查询，增加，修改和删除的功能。

       mysql>grant select, insert, update, delete on \*.\* to test2@'%' identified by '12345'；

例：增加一个test3用户，密码为12345，只能在172.21.78.89上登录，并对mydb数据库有查询，增加，修改和删除的功能。

      mysql>grant select, insert, update, delete on mydb.\* to test3@172.21.78.89 identified by '12345';

例：授权用户test4拥有数据库mydb的所有权限

      mysql>grant all privileges on mydb.\* to [test4@localhost](mailto:test3@localhost) identified by '12345';

例：创建一个只允许从本地登录的超级用户test5，并允许将权限赋予别的用户，密码为12345

      mysql>GRANT ALL PRIVILEGES ON \*.\* TO test5@localhost IDENTIFIED BY '12345' WITH GRANT OPTION;

例：创建一个一般应用程序用户，并只需要SELECT, INSERT, UPDATE, DELETE, CREATE TEMPORARY TABLES等权限，如有存储过程还需要加上EXECUTE权限。指定登录网段172.21.78网段。

        mysql>GRANT  USAGE,SELECT, INSERT, UPDATE, DELETE, SHOW VIEW ,CREATE TEMPORARY TABLES,EXECUTE ON mydb.\* TO test6@'172.21.78.%' IDENTIFIED BY  '12345';

例：创建一个普通用户(仅有查询权限)

        mysql>GRANT USAGE,SELECT ON mydb.\* TO test7@'172.21.78.%' IDENTIFIED BY  '12345';

3．查看用户权限

        查看服务器库中所有用户列表（因为用户信息保存在mysql.user表中，故可从该表中查看）

mysql> select distinct concat('user:',user,'@',host,':') as query from mysql.user;

+---------------------------+

| query                     |

+---------------------------+

| user:test1@%:             |

| user:test2@%:             |

| user:test3@172.21.78.89: |

| user:test4@172.21.78.%:  |

| user:test5@172.21.78.89:  |

| user:root@localhost:      |

| user:test1@localhost:     |

| user:test2@localhost:     |

+---------------------------+

8 rows in set (0.59 sec)

或者使用 select \* from mysql.user 可看到相关权限等信息。

查看当前登陆用户

mysql> select user();

+----------------+

| user()         |

+----------------+

| test2@localhost |

+----------------+

1 row in set (0.00 sec)

    USER()或SYSTEM\_USER()   返回当前登陆用户名

    SESSION\_USER()                     和 USER() 具有相同的意义

    SYSTEM\_USER()                      和 USER() 具有相同的意义

    CURRENT\_USER, CURRENT\_USER()    返回当前会话被验证的用户名和主机名组合

mysql> select current\_user();

+----------------+

| current\_user() |

+----------------+

| test2@localhost |

+----------------+

1 row in set (0.00 sec)

查看权限：

             SHOW GRANTS 语句： SHOWGRANTSFOR 用户名@来源地址

例：mysql> show grants for test1;

+------------------------------------------------------------------------------------------------------------------------------+

| Grants for test1@%                                                |

+------------------------------------------------------------------------------------------------------------------------------+

| GRANT SELECT, INSERT, UPDATE, DELETE ON \*.\* TO 'test1'@'%' IDENTIFIED BY PASSWORD '\*6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9' |

+------------------------------------------------------------------------------------------------------------------------------+

1 row in set (0.00 sec)

mysql> show grants for test2;

+------------------------------------------------------------------------------------------------------+

| Grants for test2@%                       |

+------------------------------------------------------------------------------------------------------+

| GRANT USAGE ON \*.\* TO 'test2'@'%' IDENTIFIED BY PASSWORD '\*00A51F3F48415C7D4E8908980D443C29C69B60C9' |

+------------------------------------------------------------------------------------------------------+

1 row in set (0.00 sec)

例：查看用户test5从主机172.21.78.89访问数据库时得授权信息。

mysql> show grants for test5@172.21.78.89;

+-----------------------------------------------------------------------------------------------------------------+

| Grants for test5@172.21.78.89                                  |

+-----------------------------------------------------------------------------------------------------------------+

| GRANT USAGE ON \*.\* TO 'test5'@'172.21.78.89' IDENTIFIED BY PASSWORD '\*00A51F3F

48415C7D4E8908980D443C29C69B60C9' |

+-----------------------------------------------------------------------------------------------------------------+

1 row in set (0.00 sec)

注意：使用show grants for test5查看不到相关信息。

mysql> show grants for test5;

ERROR 1141 (42000): There is no such grant defined for user 'test5' on host '%'

4．撤销权限：

        REVOKE 语句：用于撤销指定用户的数据库权限，撤销权限后的用户仍然可以连接到mysql服务器，但将禁止执行对应的数据库操作：

        格式：REVOKE权限列表ON 数据库名.表名FROM用户名@来源地址

        如：撤销用户test5从172.21.78.89访问数据库的所有权限：

mysql> revoke all on \*.\* from test5@172.21.78.89;

Query OK, 0 rows affected (0.00 sec)

5．刷新权限

        当mysqld服务器启动时，将授权表的内容读入到内存中。你可以通过FLUSH PRIVILEGES语句或执行**mysqladmin flush-privileges**或**mysqladmin reload**命令让它重新读取表。

        Flush privileges使权限生效，尤其在对那些权限表user、db、host等做了update或者delete更新的时候。

           mysql>FLUSH PRIVILEGES;

6．修改用户密码

        可以使用mysqladmin工具，格式：mysqladmin -u用户名 -p旧密码 password 新密码

例：给root加个密码ab12。首先在DOS下进入目录mysqlbin，然后键入以下命令：

             mysqladmin -uroot -password 12345

注：因为开始时root没有密码，所以-p旧密码一项就可以省略了。

再将root的密码改为abcde。

             mysqladmin -uroot -12345 password abcde

例：     mysqladmin -utest2 -p12345 password public12

注意，需要相关的权限

也可使用update语句更新：

     mysql>update mysql.user set password=password('123456') where User='test1' and Host='localhost';

     mysql>flush privileges;

还可使用SET PASSWORD命令，

     mysql> SET PASSWORD FOR 'USERNAME'@'HOST'=PASSWORD('NEW\_PASSWORD'

    如修改test1 密码

mysql> set password for test1@localhost = password('public12');

Query OK, 0 rows affected (0.00 sec)

7．删除用户

        mysql>delete from user where user='test2' and host='localhost';

        mysql>flush privileges;

        注意删除用户最好不要使用DELETE直接删除，因为使用DELETE删除后用户的权限并未删除，新建同名用户后又会继承以前的权限。

        正确的做法是使用DROP USER命令删除用户，比如要删除'test'@'172.168.100.%'用户采用如下命令：

        mysql>DROP USER test3@'172.21.78.89’;  ---- 删除账户及权限