## 密码

为了加强安全性，MySQL5.7为root用户随机生成了一个密码，在error log中，关于error log的位置，如果安装的是RPM包，则默认是/var/log/mysqld.log。

一般可通过log\_error设置

mysql> select **@@log\_error**;

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

| **@@log\_error** |

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

| /var/log/mysqld.log |

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

**1** row in set (**0.00** sec)

可通过# grep "password" /var/log/mysqld.log 命令获取MySQL的临时密码

**2016**-**01**-19T05:**16**:**36**.218234Z **1** [Note] A temporary password is generated for root@localhost: waQ,qR%be2(**5**

用该密码登录到服务端后，必须马上修改密码，不然会报如下错误：

mysql> select user();

ERROR **1820** (HY000): You must reset your password using ALTER USER statement before executing this statement.

如果只是修改为一个简单的密码，会报以下错误：

mysql> ALTER USER USER() IDENTIFIED BY '12345678';

ERROR **1819** (HY000): Your password does not satisfy the current policy requirements

这个其实与validate\_password\_policy的值有关。

validate\_password\_policy有以下取值：

| **Policy** | **Tests Performed** |
| --- | --- |
| 0 or LOW | Length |
| 1 or MEDIUM | Length; numeric, lowercase/uppercase, and special characters |
| 2 or STRONG | Length; numeric, lowercase/uppercase, and special characters; dictionary file |

默认是1，即MEDIUM，所以刚开始设置的密码必须符合长度，且必须含有数字，小写或大写字母，特殊字符。

有时候，只是为了自己测试，不想密码设置得那么复杂，譬如说，我只想设置root的密码为123456。

必须修改两个全局参数：

首先，修改validate\_password\_policy参数的值

mysql> set global validate\_password\_policy=**0**;

Query OK, **0** rows affected (**0.00** sec)

这样，判断密码的标准就基于密码的长度了。这个由validate\_password\_length参数来决定。

mysql> select **@@validate\_password\_length**;

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

| **@@validate\_password\_length** |

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

| **8** |

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

**1** row in set (**0.00** sec)

validate\_password\_length参数默认为8，它有最小值的限制，最小值为：

validate\_password\_number\_count

+ validate\_password\_special\_char\_count

+ (**2** \* validate\_password\_mixed\_case\_count)

其中，validate\_password\_number\_count指定了密码中数据的长度，validate\_password\_special\_char\_count指定了密码中特殊字符的长度，validate\_password\_mixed\_case\_count指定了密码中大小字母的长度。

这些参数，默认值均为1，所以validate\_password\_length最小值为4，如果你显性指定validate\_password\_length的值小于4，尽管不会报错，但validate\_password\_length的值将设为4。如下所示：

mysql> select **@@validate\_password\_length**;

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

| **@@validate\_password\_length** |

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

| **8** |

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

**1** row in set (**0.00** sec)

mysql> set global validate\_password\_length=**1**;

Query OK, **0** rows affected (**0.00** sec)

mysql> select **@@validate\_password\_length**;

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

| **@@validate\_password\_length** |

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

| **4** |

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

**1** row in set (**0.00** sec)

如果修改了validate\_password\_number\_count，validate\_password\_special\_char\_count，validate\_password\_mixed\_case\_count中任何一个值，则validate\_password\_length将进行动态修改。

mysql> select **@@validate\_password\_length**;

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

| **@@validate\_password\_length** |

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

| **4** |

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

**1** row in set (**0.00** sec)

mysql> select **@@validate\_password\_mixed\_case\_count**;

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

| **@@validate\_password\_mixed\_case\_count** |

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

| **1** |

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

**1** row in set (**0.00** sec)

mysql> set global validate\_password\_mixed\_case\_count=**2**;

Query OK, **0** rows affected (**0.00** sec)

mysql> select **@@validate\_password\_mixed\_case\_count**;

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

| **@@validate\_password\_mixed\_case\_count** |

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

| **2** |

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

**1** row in set (**0.00** sec)

mysql> select **@@validate\_password\_length**;

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

| **@@validate\_password\_length** |

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

| **6** |

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

**1** row in set (**0.00** sec)

当然，前提是validate\_password插件必须已经安装，MySQL5.7是默认安装的。

那么如何验证validate\_password插件是否安装呢？可通过查看以下参数，如果没有安装，则输出将为空。

mysql> SHOW VARIABLES LIKE 'validate\_password%';

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

| Variable\_name | Value |

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

| validate\_password\_dictionary\_file | |

| validate\_password\_length | **6** |

| validate\_password\_mixed\_case\_count | **2** |

| validate\_password\_number\_count | **1** |

| validate\_password\_policy | LOW |

| validate\_password\_special\_char\_count | **1** |

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

**6** rows in set (**0.00** sec)

## 添加用户

**创建用户:**

**命令**:

CREATE USER 'username'@'host' IDENTIFIED BY 'password';

说明:username – 你将创建的用户名, host – 指定该用户在哪个主机上可以登陆,如果是本地用户可用localhost,  如 果想让该用户可以从任意远程主机登陆,可以使用通配符%. password –  该用户的登陆密码,密码可以为空,如果为空则该用户可以不需要密码登 陆服务器.

**例子：**

CREATE USER 'dog'@'localhost' IDENTIFIED BY '123456';

CREATE USER 'pig'@'192.168.1.101\_' IDENDIFIED BY '123456';

CREATE USER 'pig'@'%' IDENTIFIED BY '123456';

CREATE USER 'pig'@'%' IDENTIFIED BY '';

CREATE USER 'pig'@'%';

### 用户授权:

**命令:**

GRANT privileges ON databasename.tablename TO 'username'@'host'

说明: privileges – 用户的操作权限,如SELECT , INSERT , UPDATE  等(详细列表见该文最后面).如果要授予所 的权限则使用ALL.;databasename –  数据库名,tablename-表名,如果要授予该用户对所有数据库和表的相应操作权限则可用\* 表示, 如\*.\*.  
例子:

GRANT SELECT, INSERT ON test.user TO 'pig'@'%';

GRANT ALL ON \*.\* TO 'pig'@'%';

注意:用以上命令授权的用户不能给其它用户授权,如果想让该用户可以授权,用以下命令:

GRANT privileges ON databasename.tablename TO 'username'@'host' WITH GRANT OPTION；

设置与更改用户密码

命令:

SET PASSWORD FOR 'username'@'host' = PASSWORD('newpassword');

如果是当前登陆用户用

SET PASSWORD = PASSWORD("newpassword");

撤销用户权限

命令:

REVOKE privilege ON databasename.tablename FROM 'username'@'host';

说明: privilege, databasename, tablename – 同授权部分.  
例子

REVOKE SELECT ON \*.\* FROM 'pig'@'%';

注意: 假如你在给用户’pig’@'%’授权的时候是这样的(或类似 的):GRANT SELECT ON test.user TO  ‘pig’@'%’, 则在使用 REVOKE SELECT ON \*.\* FROM  ‘pig’@'%’;命令并不能撤销该用户对test数据库中user表的SELECT 操作. 相反,如果授权使用的是GRANT SELECT ON  \*.\* TO ‘pig’@'%’;则 REVOKE SELECT ON test.user FROM  ‘pig’@'%’;命令也不能撤销该用户对test数据库中user表的 Select 权限.

具体信息可以用命令SHOW GRANTS FOR ‘pig’@'%’; 查看.

删除用户

命令:

DROP USER ‘username’@'host’;

一个典型的数据库建表, 建用户过程:

创建用于localhost连接的用户并指定密码

mysql> create user 'pcom'@'localhost' identified by 'aaa7B2249';

Query OK, 0 rows affected (0.00 sec)

创建数据库

mysql> create database pcom default character set utf8 collate utf8\_bin;

Query OK, 1 row affected (0.00 sec)

给本地用户授权, 这里不需要指定密码

mysql> grant all on pcom.\* to 'pcom'@'localhost';

Query OK, 0 rows affected (0.00 sec)

给其他IP地址下的用户授权, 注意: 这里必须指定密码, 否则就可以无密码访问

mysql> grant all on pcom.\* to 'pcom'@'192.168.0.0/255.255.0.0' identified by 'aaa7B2249';

Query OK, 0 rows affected (0.00 sec)

同理

mysql> grant all on pcom.\* to 'pcom'@'172.20.0.0/255.255.0.0' identified by 'aaa7B2249';

Query OK, 0 rows affected (0.00 sec)

Done!

附表:在MySQL中的操作权限

ALTER

Allows use of ALTER TABLE.

ALTER ROUTINE

Alters or drops stored routines.

CREATE

Allows use of CREATE TABLE.

CREATE ROUTINE

Creates stored routines.

CREATE TEMPORARY TABLE

Allows use of CREATE TEMPORARY TABLE.

CREATE USER

Allows use of CREATE USER, DROP USER, RENAME USER, and REVOKE ALL PRIVILEGES.

CREATE VIEW

Allows use of CREATE VIEW.

DELETE

Allows use of DELETE.

DROP

Allows use of DROP TABLE.

EXECUTE

Allows the user to run stored routines.

FILE

Allows use of SELECT… INTO OUTFILE and LOAD DATA INFILE.

INDEX

Allows use of CREATE INDEX and DROP INDEX.

INSERT

Allows use of INSERT.

LOCK TABLES

Allows use of LOCK TABLES on tables for which the user also has SELECT privileges.

PROCESS

Allows use of SHOW FULL PROCESSLIST.

RELOAD

Allows use of FLUSH.

REPLICATION

Allows the user to ask where slave or master

CLIENT

servers are.

REPLICATION SLAVE

Needed for replication slaves.

SELECT

Allows use of SELECT.

SHOW DATABASES

Allows use of SHOW DATABASES.

SHOW VIEW

Allows use of SHOW CREATE VIEW.

SHUTDOWN

Allows use of mysqladmin shutdown.

SUPER

Allows use of CHANGE MASTER, KILL, PURGE MASTER LOGS, and SET GLOBAL SQL statements. Allows mysqladmin debug command. Allows one extra connection to be made if maximum connections are reached.

UPDATE

Allows use of UPDATE.

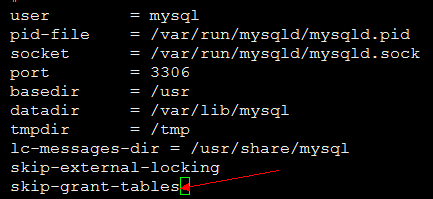
USAGE

Allows connection without any specific privileges.

最后不要忘记刷新数据库

flush privileges;

## Mysql root密码忘记



http://www.linuxidc.com/Linux/2012-12/75231.htm

update user set password=PASSWORD("12345678") where user='root';

## 添加用户报错

mysql> CREATE USER 'opensips'@'%' IDENTIFIED BY '12345678';

ERROR 1396 (HY000): Operation CREATE USER failed for 'opensips'@'%'

mysql> use mysql;

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

mysql> select user from user;

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

| user |

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

| opensips |

| test |

| root |

| root |

| debian-sys-maint |

| opensips |

| root |

| root |

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

8 rows in set (0.01 sec

查看该用户是否已存在

## [查看mysql数据库及表编码格式](http://www.cnblogs.com/shootercheng/p/5836657.html)

1.查看数据库编码格式

|  |  |
| --- | --- |
| 1 | mysql> show variables like 'character\_set\_database'; |

 2.查看数据表的编码格式

|  |  |
| --- | --- |
| 1 | mysql> show create table <表名>; |

 3.创建数据库时指定数据库的字符集

mysql>create database <数据库名> character set utf8;

4.创建数据表时指定数据表的编码格式

create table tb\_books (

name varchar(**45**) not null,

price double not null,

bookCount int not null,

author varchar(**45**) not null ) default charset = utf8;

5.修改数据库的编码格式

mysql>alter database <数据库名> character set utf8;

6.修改数据表格编码格式

mysql>alter table <表名> default character set utf8;

7.修改字段编码格式

mysql>alter table <表名> change <字段名> <字段名> <类型> character set utf8;

mysql>alter table user change username username varchar(**20**) character set utf8 not null;

 8.添加外键

mysql>alter table tb\_product add constraint fk\_1 foreign key(factoryid) references tb\_factory(factoryid);

mysql>alter table <表名> add constraint <外键名> foreign key<字段名> REFERENCES <外表表名><字段名>;

9.删除外键

mysql>alter table tb\_people drop foreign key fk\_1;

mysql>alter table <表名> drop foreign key <外键名>;