

# MySQL密码问题

我们在平时的生产以及实验环境中，经常会遇到很多关于密码的问题，这里我们以MySQL密码为例来简单的解析一下密码的安全以及对于密码的操作。

案例一：

mysql的root密码被遗忘了，我应该怎么操作才是最安全的？

答：编辑配置文件/etc/my.cnf 在其中添加skip-grant-tables 重新启动mysql服务 命令行输入 mysql -uroot 直接登陆，进入数据库后修改数据库密码

```
> UPDATE mysql.user SET password=password("你的密码") WHERE user=root;

> flush privileges;

$ sudo vim /etc/my.cnf 删除skip-grant-tables

$ sudo systemctl restart mysqld
```

加入skip-grant-tables的意思是启动mysql服务的时候跳过权限表认证 启动后，连接到mysql的root不需要口令

案例二：

mysql5.7在安装后找不到初始密码？该怎么办？

答：参照案例一方法修改密码

答：使用 `mysqld --initialize` 命令初始化数据库，然后查看/etc/my.cnf 确定mysql.log日志位置

使用命令 `> grep 'temporary password' /var/log/mysqld.log` 查看初始密码

案例三：

怎么样才能让mysql密码验证策略变得简单一些？

查看现有的密码策略

```
mysql> SHOW VARIABLES LIKE 'validate_password%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| validate_password_dictionary_file |      |
| validate_password_length | 8     |
| validate_password_mixed_case_count | 1     |
| validate_password_number_count | 1     |
```

validate_password_policy	MEDIUM
validate_password_special_char_count	1

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

validate\_password\_number\_count参数是密码中至少含有的数字个数，当密码策略是MEDIUM或以上时生效。

validate\_password\_special\_char\_count参数是密码中非英文数字等特殊字符的个数，当密码策略是MEDIUM或以上时生效。

validate\_password\_mixed\_case\_count参数是密码中英文字符大小写的个数，当密码策略是MEDIUM或以上时生效。

validate\_password\_length参数是密码的长度，这个参数由下面的公式生成

validate\_password\_number\_count+ validate\_password\_special\_char\_count+ (2 \* validate\_password\_mixed\_case\_count)

validate\_password\_dictionary\_file参数是指定密码验证的字典文件路径。

validate\_password\_policy这个参数可以设为0、1、2，分别代表从低到高的密码强度，此参数的默认值为1，如果想将密码强度改若，则更改此参数为0。

创建用户时报错：

```
mysql> CREATE USER 'test'@'localhost' IDENTIFIED BY 'test';
ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
```

报错原因：

指定的密码没有符合现有的密码策略。

解决方法：

可以按照现有策略设置密码，也可以更改密码策略。

① 按照现有密码策略指定密码

```
mysql> CREATE USER 'test'@'localhost' IDENTIFIED BY 'System#2016';
Query OK, 0 rows affected (0.16 sec)
```

② 更改密码策略，降低密码的验证标准

--更改密码策略为LOW

```
mysql> set global validate_password_policy=0;
Query OK, 0 rows affected (0.00 sec)
```

--更改密码长度

```
mysql> set global validate_password_length=0;
Query OK, 0 rows affected (0.00 sec)
```

--密码最小长度为4

```
mysql> SHOW VARIABLES LIKE 'validate_password%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
```

validate_password_dictionary_file	
validate_password_length	4
validate_password_mixed_case_count	1
validate_password_number_count	1
validate_password_policy	LOW
validate_password_special_char_count	1

6 rows in set (0.00 sec)

```
mysql> drop user 'test'@localhost;
Query OK, 0 rows affected (0.07 sec)
```

--创建长度为3的密码报错

```
mysql> CREATE USER 'test'@'localhost' IDENTIFIED BY 'tes';
ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
```

--创建长度为4的密码，4为现有密码策略的最小长度

```
mysql> CREATE USER 'test'@'localhost' IDENTIFIED BY 'test';
Query OK, 0 rows affected (0.01 sec)
```

最后一点需要注意的是：mysql在登录时，是不建议在命令行中输入密码的，一定要密文输入，这也是为了保证密码不会在历史记录中出现，防止密码的外泄！