



链滴

MySQL8.0 密码管理与密码策略

作者: [luomuren](#)

原文链接: <https://ld246.com/article/1648172380160>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

MySQL8.0 密码管理与密码策略

说明

MySQL 的版本在 8.0 以上。本文使用的版本是 8.0.13 这个版本。

```
mysql> select version();
+-----+
| version() |
+-----+
| 8.0.13   |
+-----+
1 row in set
```

密码管理

在 mysql 8.0 中新增了三个变量，用于对密码的管理：

```
## 新密码不能和前面三次的密码相同
```

```
password_history = 3 ;
```

```
## 新密码不能和前面九十天内使用的密码相同
```

```
password_reuse_interval = 90 ;
```

```
## 默认为off；为on时 修改密码需要用户提供当前密码 (开启后修改密码需要验证旧密码，root 用不需要)
```

```
password_require_current = on ;
```

查看密码管理策略

```
mysql> show variables like 'password%';
+-----+-----+
| Variable_name      | Value |
+-----+-----+
| password_history   | 0     |
| password_require_current | OFF  |
| password_reuse_interval | 0     |
+-----+-----+
3 rows in set
```

对密码相关参数的修改

全局设置

第一种方式：修改 mysql 的配置文件，全局生效

```
## 找到mysql的配置文件 my.cnf, 我的在 /etc/my.cnf
```

vi /etc/my.cnf

在最后增加一行，保存退出

password_history=6

```
# Remove leading # to set options mainly useful for reporting servers.
# The server defaults are faster for transactions and fast SELECTs.
# Adjust sizes as needed, experiment to find the optimal values.
# join_buffer_size = 128M
# sort_buffer_size = 2M
# read_rnd_buffer_size = 2M
#
# Remove leading # to revert to previous value for default_authentication_plugin,
# this will increase compatibility with older clients. For background, see:
# https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_default_authentication_plugin
# default_authentication_plugin=mysql_native_password

datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock

log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
password_history=3
"/etc/my.cnf" 32L, 1262C written
```

https://blog.csdn.net/qq_18948359

第二种方式:

使用命令 `set persist password_history=6;` 这个具体的实现是增加了一个配置文件

```
mysql80 > set persist password_history=6;
Query OK, 0 rows affected (0.00 sec)

mysql80 > show variables like 'password%';
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| password_history       | 6     |
| password_require_current | OFF   |
| password_reuse_interval | 0     |
+-----+-----+
3 rows in set (0.00 sec)

mysql80 > exit
Bye
```

https://blog.csdn.net/qq_18948359

查看新增的配置文件

```
[root@mysql80 ~]# more /var/lib/mysql/mysqld-auto.cnf
{ "Version" : 1 , "mysql_server" : { "password_history" : { "Value" : "6" , "Metadata" :
{ "Timestamp" : 1545233453366962 , "User" : "root" , "Host" : "localhost" } } } }
[root@mysql80 ~]#
```

对指定用户设置

先查询看下，发现 Password_reuse_history 这参数都是为 null，就表示使用全局配置的参数

mysql> select host, user, Password_reuse_history from mysql.user;

```
+-----+-----+-----+
```

```

| host      | user          | Password_reuse_history |
+-----+-----+-----+
| localhost | mysql.infoschema | NULL                    |
| localhost | mysql.session   | NULL                    |
| localhost | mysql.sys       | NULL                    |
| localhost | root           | NULL                    |
+-----+-----+-----+
4 rows in set

```

我是为了方便与测试，大家不要瞎改 root 用户的权限啊

```
mysql> ALTER USER 'root'@'localhost' password history 3;
Query OK, 0 rows affected
```

这里重新查询一次，发现已经修改了

```
mysql> select host, user, Password_reuse_history from mysql.user;
+-----+-----+-----+
| host      | user          | Password_reuse_history |
+-----+-----+-----+
| localhost | mysql.infoschema | NULL                    |
| localhost | mysql.session   | NULL                    |
| localhost | mysql.sys       | NULL                    |
| localhost | root           | 3                       |
+-----+-----+-----+
4 rows in set

```

接下来，我们修改密码看下效果，会提示错误，不让修改

```
mysql> alter user 'root'@'localhost' identified by 'root@123456';
1819 - Your password does not satisfy the current policy requirements
mysql>
```

密码策略

关于 mysql 密码策略相关参数：

validate_password.length 固定密码的总长度；
 validate_password.dictionary_file 指定密码验证的文件路径；
 validate_password.mixed_case_count 整个密码中至少要包含大/小写字母的总个数；
 validate_password.number_count 整个密码中至少要包含阿拉伯数字的个数；
 validate_password.special_char_count 整个密码中至少要包含特殊字符的个数；
 validate_password.policy 指定密码的强度验证等级，默认为 MEDIUM；

关于 validate_password.policy 的取值：

0/LOW：只验证长度；
 1/MEDIUM：验证长度、数字、大小写、特殊字符；
 2/STRONG：验证长度、数字、大小写、特殊字符、字典文件；

查看当前的密码策略

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

```

```

| Variable_name          | Value |
+-----+-----+
| validate_password.check_user_name | ON    |
| 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    |
+-----+-----+
7 rows in set

```

修改密码策略

首先需要设置密码的验证强度等级，设置 `validate_password_policy` 的全局参数为 `LOW` 即可

```

set global validate_password.policy=LOW;
## 修改密码长度
set global validate_password.length=6;

```

```

mysql> set global validate_password.policy=LOW;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW VARIABLES LIKE 'validate_password%':
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| validate_password.check_user_name | ON    |
| validate_password.dictionary_file |      |
| validate_password.length          | 8     |
| validate_password.mixed_case_count | 1     |
| validate_password.number_count    | 1     |
| validate_password.policy          | LOW   |
| validate_password.special_char_count | 1    |
+-----+-----+
7 rows in set (0.01 sec)

mysql> set global validate_password.length=6;
Query OK, 0 rows affected (0.00 sec)

mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.02 sec)

```

版权声明

作者: [Wayfreem]

出处: https://blog.csdn.net/qq_18948359/article/details/105837540

未经作者同意必须保留此段声明，且在文章页面明显位置给出原文链接，否则保留追究法律责任的权利。