

# Mariadb 数据库服务器

RHEL7开始数据库服务器取消了传统的mysql，改由mariadb替代。Mysql被oracle收购，导致红帽认为未来有可能甲骨文会将mysql闭源。mariadb和mysql出自同一个团队，其实在操作方面也几乎是一模一样的。

```
[root@rhel7 mnt]# yum install mariadb* -y    安装数据库
```

```
[root@rhel7 mnt]# systemctl start mariadb    启动服务
```

```
[root@test ~]# mysql_secure_installation
```

```
[root@rhel7 mnt]# firewall-cmd --add-service=mysql
```

```
[root@rhel7 mnt]# firewall-cmd --add-service=mysql --permanent
```

数据库基本操作：

```
[root@rhel7 ~]# mysql
```

```
MariaDB [(none)]> select version(); 查看版本信息
```

```
MariaDB [(none)]> select user(); 查看用户
```

```
MariaDB [mysql] > select user();
```

```
+-----+
| user() |
+-----+
```

```
| root@localhost |
+-----+
```

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

```
# 当前为root用户，本地登录MariaDB数据库
```

```
MariaDB [(none)]> status;    查看状态
```

```
MariaDB [(none)]> system ls    在mariadb下使用系统的shell指令,system是关键词，表示使用系统指令，也可以用\!替代system
```

```
MariaDB [(none)]> create database rhel7;    新建一个名叫rhel7的数据库
```

```
MariaDB [(none)]> show databases; # 查看当前的数据库
```

```
MariaDB [(none)]> drop database rhel7;    删除数据库
```

```
MariaDB [(none)]> create database rhel7 charset=gbk;    创建数据库时指定编码
```

```
MariaDB [(none)]> alter database rhel7 charset=utf8;    更改已经存在的数据库的编码
```

```
MariaDB [(none)]> use rhel7;    切换到rhel7数据库中
```

```
MariaDB [rhel7]> show tables;    查看数据库中的表
```

示例：生成第一张表

```
MariaDB [rhel7]> create table xx (
```

```
-> id int,
```

```
-> name varchar(10),
```

```
-> age int,
```

```
-> phone int(11)
```

```
-> );
```

```
MariaDB [rhel7]> desc xx;    查看表xx的结构
```

```
MariaDB [rhel7]> insert into xx    向xx表中插入内容
```

```
-> (id,name,age,phone)    插入的具体的值的属性
```

```
-> value('1','zs','25','1398888888');    插入的具体的某一行的值
```

```
MariaDB [rhel7]> select * from xx;
```

```
MariaDB [rhel7]> select name from xx;
```

```
MariaDB [rhel7]> select name,age from xx;
```

```
MariaDB [rhel7]> select * from xx where age=22;
```

```
MariaDB [rhel7]> select * from xx where name='ls';    根据特定的值去查找表中相应的行的内容
```

```
MariaDB [rhel7]> rename table xx to yy;    更改表的名称
```

```
MariaDB [rhel7]> alter table yy drop phone;    删除表中的列
```

```
MariaDB [rhel7]> alter table yy add mail varchar(20) after phone;    添加新列到表中
```

```
MariaDB [rhel7]> select phone from xx where id=(select id from yy where name='ww');
```

数据库的安全：

设置mysql登陆密码

```
MariaDB [mysql]> update mysql.user set password=password('china') where user='root' and host='localhost'    指明针对mysql数据库的user表的哪一行进行修改，这里是指定的user为root，host为localhost的内容的password进行修改。
```

注：修改成功后，重新登陆数据库仍然是使用的以前的旧的密码，是因为服务器在不重启的情况下，缓存中会保留以前的密码信息，所以新密码需要在缓存中也进行定义，或者直接重启服务器

```
MariaDB [(none)]> flush privileges;    刷新缓存
```

重置数据库密码：

方法一：MariaDB [mysql]> **update mysql.user set password=null;** 将空值写入mysql.user中

MariaDB [(none)]> **flush privileges;** 刷新缓存

方法二：如果根本无法进入数据库，那么第一种方法是无法使用的，这样，我们就必须修改配置文件来达到重置密码的目的。前提是数据库是未启动的。

[root@localhost ~]# **systemctl stop mariadb** 停止数据库服务的运行

[root@localhost ~]# **vim /etc/my.cnf** 修改数据库主配置文件

```
1 [mysqld]
2 datadir=/var/lib/mysql
3 socket=/var/lib/mysql/mysql.sock
4 # Disabling symbolic-links is recommended to prevent assorted security
5 symbolic-links=0
6 skip-grant-tables
7 # Settings user and group are ignored when systemd is used.
8 # If you need to run mysqld under a different user or group,
```

[root@localhost ~]# **systemctl restart mariadb** 重启数据库服务并进入mysql数据库重置密码

MariaDB [(none)]> **update mysql.user set password=password('redhat') where host='localhost' and user='root';**

MariaDB [(none)]> **flush privileges;**

方法三：通过命令行的方式修改

[root@localhost ~]# **systemctl stop mariadb**

[root@localhost ~]# **mysqld\_safe --skip-grant-tables** 通过命令指定进入安全模式

MariaDB [(none)]> **update mysql.user set password=password('redhat') where host='localhost' and user='root';**

MariaDB [(none)]> **flush privileges;**

[root@localhost ~]# **systemctl start mariadb**

### 数据库的账号管理：

MariaDB [mysql]> **create user tom;** 创建一个用户tom，通过这种方式创建的用户可以用于任何主机的登陆

MariaDB [mysql]> **create user jarry@localhost;** 通过这种方式创建的用户只能在本地上服务器上登陆

MariaDB [mysql]> **create user xxx@0.0.0.0 identified by "admin";**

MariaDB [mysql]> **select Host,User,Password FROM mysql.user;** 对比创建的账户，我们发现默认情况下用create创建的用户是没有密码

如果希望创建用户可以用来登陆数据库，我们还需要使用grant命令来设置新建用户的权限和密码

### grant语法：

**grant 权限1 权限2 .. on 数据库.表 to 用户@主机名 identified by 'password'**

//为主机192.168.24.252的用户tom设置登陆数据库的权限，权限是create，密码是redhat

MariaDB [mysql]> **grant create on rhel7.xx to tom@'192.168.24.252' identified by 'redhat';**

MariaDB [mysql]> **show grants for tom@192.168.24.252\G** 查看指定用户的所有权限

MariaDB [mysql]> **grant drop on rhel7.xx to tom@'192.168.24.252';** 添加用户权限

MariaDB [mysql]> **revoke drop on rhel7.xx from tom@'192.168.24.252';** 回收用户权限

MariaDB [rhel7]> **drop tables xx;** tom用户登录后再次尝试删除失败

### 数据库的备份和恢复

#### a. 单表的恢复：

[root@localhost ~]# **mysqldump -uroot -p rhel7 xx** 使用root身份备份rhel7库中的表xx，最终得到的结果是输出在屏幕上的文件。可以直接使用重定向来实现备份。

在数据库中删除rhel7库中的xx表，然后用下面的方式恢复

[root@localhost tmp]# **mysqldump -uroot -p rhel7 xx > rhel.xx.backup**

MariaDB [rhel7]> **source /root/rhel.xx.backup;** 恢复rhel7.xx

#### b. 数据库的恢复：

方式一：

[root@localhost ~]# **mysqldump -uroot -p rhel7 > rhel7.backup** 备份数据库rhel7中所有的表。因为备份的其实是表，所以，如果库不存在，表是无法恢复的。

MariaDB [rhel7]> **drop database rhel7;** 删除整个库，

MariaDB [(none)]> **source /root/rhel7.backup** 恢复数据库，失败。因为数据库不存在，表无法导入

MariaDB [(none)]> **create database rhel7;** 为了恢复成功，需要首先创建数据库

MariaDB [(rehl7)]> **source /root/rhel7.backup** 再次执行导入恢复

方式二：

[root@localhost ~]# **mysqldump -uroot -p -B rhel7 > rhel7.bu** 参数B表示备份整个库，这个命令可以直接将整个库做备份，这种方式不依赖本身是否数据库是否存在

MariaDB [(none)]> **source /root/rhel7.bu**