

### 一 用户授权

#### 1.1 用户授权:

在数据库服务器上添加客户端连接时使用的用户,并设置访问权限

原因:默认只允许数据库管理员 root 用户本机登录

**which mysql:** 查看 mysql 是否安装,若安装显示安装路径

**mysql> select password("密码");** 查看密码的加密字符串

#### 1.2 命令格式

**grant 权限列表 on 库.表 to 用户名@"客户机地址" identified by "密码"**

**[with grant option];** #with grant option 表示被授权用户也可授权

1.2.1 权限列表	说明
<b>all</b>	所有权限
<b>usage</b>	无权限
<b>select,update,insert</b>	个别权限
<b>select,update(字段 1,...,字段 n)</b>	指定字段

1.2.2 库名	说明
<b>*.*</b>	所有库所有表
<b>库.*</b>	一个库
<b>库.表</b>	一张表

#### 1.2.3 用户名

授权时自定义,要有标识性

存储在 mysql 库的 user 表里

### 1.2.4 客户机地址表示方式

% 所有主机 192.168.4.% 网段内主机

192.168.4.1 1 台主机      localhost      数据库服务器本机

### 1.3 授权查看

**select user();**      显示可登录用户名及客户机地址

**show grants;**                    显示用户自身访问权限

**show grants for 用户名@"客户机地址";** 管理员查看已有授权用户的权限

**set password=password("密码");** 授权用户连接后修改连接密码

**set password for 用户名@"客户机地址"=password("密码");**

### 管理员重置授权用户连接密码

**drop user 用户名@"客户机地址" ;** 有权限的管理员删除授权用户

例子：

用户授权(50 51 52 ...表示主机)

```
50 mysql> grant all on db3.* to yaya@"%" identified by "123456";
```

```
mysql51 ~]# mysql -h192.168.4.50 -uyaya -p123456
```

### #给 yaya 用户授权,并在 51 上登录

51 mysql> select user(); #用户 yaya 在 51 登录后查看可登录用户及权限

-----+

```
| user() |
```

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

```
| yaya@192.168.4.51 |
```

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

```
51 mysql> show grants; #用户 yaya 在 51 上登录后查看自身权限
```

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

```
| Grants for yaya@% |
```

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

```
| GRANT USAGE ON *.* TO 'yaya'@'%' |
```

```
| GRANT ALL PRIVILEGES ON `db3`.* TO 'yaya'@'%' |
```

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

```
mysql> show databases; #用户 yaya 在 51 上登录后查看数据库
```

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

```
| Database | #information_schema 是一个假库
```

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

```
| information_schema |
```

```
| db3 |
```

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

```
51 mysql> set password=password("654321"); #修改用户 yaya 的密码
```

```
51 mysql> exit
```

```
mysql51 ~]# mysql -uyaya -p654321 -h192.168.4.50
```

#验证新密码是否能登录

```
50 mysql> set password for yaya@"%"=password("123456");
```

#管理员重置 yaya 登录密码

```
mysql51 ~]# mysql -uyaya -p123456 -h192.168.4.50
```

#验证重置后密码是否能登录

```
50 mysql> show grants for yaya@"%";
```

```
+-----+
| Grants for yaya@%                                |
+-----+
| GRANT USAGE ON *.* TO 'yaya'@'%'                  |
| GRANT ALL PRIVILEGES ON `db3`.* TO 'yaya'@'%'    |
| GRANT ALL PRIVILEGES ON `db4`.* TO 'yaya'@'%'    |
+-----+
```

#数据库管理员查看 yaya 的授权

```
50 mysql> drop user yaya@"%";    #删除用户 yaya
```

```
mysql51 ~]# mysql -uyaya -p123456 -h192.168.4.50
```

#删除用户 yaya 后在 51 尝试用 yaya 登录

## 1.4 msql 库记录授权信息表作用

**user 表**                      记录已有的授权用户权限

**db 表**                      记录已有授权用户对数据库的访问权限

**tables\_priv 表**        记录已有授权用户对表的访问权限

**cloumns\_priv 表**    记录已有授权用户对字段的访问权限

查看以上表记录可以获取用户权限;也可以通过修改更新表内的记录,修改用户权限

### 1.4.1 撤销权限

**revoke 权限列表 on 库.表 from 用户名@"客户机地址"**

**例子**

```
50 mysql> desc user; #查看 mysql.user 表结构
```

```
50 mysql> show grants; #查看数据库管理员 root 用户的权限
```

```
GRANT PROXY ON ''@'' TO 'root'@'localhost' WITH GRANT OPTION
```

    #该句表示 'root'@'localhost' 有授权权限

```
50 mysql> select * from user where user="root"and host ="localhost"\G; # 查看 user 表下 user="root" and host ="localhost" 行的其他字段的值,\G 表示以列型输出
```

```
50 mysql> grant all on *.* to root@"%" identified by "123456" with grant option; #给所有主机的 root 用户授权可以用 123456 登录数据库服务器,并对所有库.表拥有所有权限,并可以对其他用户授权
```

```
[revoke grant option on *.* from root@"%";]
```

```
mysql51 ~]# mysql -uroot -p123456 -h192.168.4.50
```

#51 上以 root 用户验证登录 50 上的数据库

51 mysql> show grants; #查看 51 上以 root 登录 50 上的数据库后的权限

51 mysql> select \* from mysql.columns\_priv;

51 mysql> update mysql.columns\_priv

set column\_priv="select,insert";

51 mysql> flush privileges;

51 mysql> select \* from mysql.columns\_priv;

## 二 root 密码

### 2.1 root 密码忘了怎么办? 恢复 root 密码

#### 2.1.1 停止 mysql 服务程序

#### 2.1.2 跳过授权表启动 mysql 服务程序

#### 2.1.3 修改 root 密码

#### 2.1.4 以正常方式重启 mysql 服务程序

### 例子

#### 知道旧密码修改新密码

```
mysql50 ~]# mysqladmin -hlocalhost -uroot -p password "456789"
```

Enter password: #输入旧密码

```
mysql50 ~]# mysql -uroot -p456789 #使用新密码登录
```

#### 忘记旧密码修改新密码

```
mysql50 ~]# systemctl stop mysqld #关闭服务
```

```
mysql50 ~]# systemctl status mysqld #确认服务关闭
```

mysql50 ~]# vim /etc/my.cnf #修改配置文件

[mysqld]

**skip-grant-tables** #添加此行

**#validate\_password\_policy=0** #注销

**#validate\_password\_length=6** #注销

**systemctl retart mysqld** #重启服务

mysql50 ~]# mysql #进入数据库

50 mysql> select \* from mysql.user where host="localhost" and

user="root"\G; #查看 mysql.user 下 user="root" host="localhost"时

**authentication\_string** 字段的值

50 mysql> **update mysql.user set authentication\_string=passwo**

**rd("123abc") where user="root" and host="localhost";** #重置密码

mysql> flush privileges; #刷新

mysql50 ~]# vim /etc/my.cnf #修改配置文件

[mysqld]

**#skip-grant-tables** #注销

**validate\_password\_policy=0** #开启

**validate\_password\_length=6** #开启

mysql50 ~]# systemctl restart mysqld #重启服务

```
mysql50 ~]# mysql -uroot -p123abc #用密码登录
```

### 三 备份

**3.1 物理备份**(缺陷:要求系统都为同一类型;完全备份,数据量比较大)

冷备:cp tar ...

#### 3.1.1 创建备份文件

```
cp -r /var/lib/mysql /备份目录路径/mysql.bak #本机创建备份文件
```

```
tar -zcvf /root/mysql.tar.gz /var/lib/mysql/* #打包备份文件
```

#### 3.1.2 关闭服务

```
systemctl stop mysqld
```

#### 3.1.3 将备份文件放入数据库目录

```
cp -r /备份路径/mysql.bak /var/lib/mysql/
```

```
tar -xzvf /root/mysql.tar.gz -C /var/lib/mysql/
```

#### 3.1.4 修改数据库目录内文件所属

```
chown -R mysql:mysql /var/lib/mysql
```

#### 3.1.5 重启服务并检查进程和端口

```
systemctl restart mysqld
```

```
ps -C mysqld #查看进程
```

```
ss -antulp #查看端口
```

#### 3.1.6 最后登录 mysql 查看数据库,数据表

### 例子



## 物理备份

```
mysql50 ~]# cp -r /var/lib/mysql /root/mysql.bak
```

#创建 1 个备份文档

```
mysql50 ~]# ls -ld /root/mysql.bak
```

```
drwxr-x--x 7 root root 4096 9月  2 15:27 /root/mysql.bak
```

```
mysql50 ~]# tar -zcvf /root/mysql.tar.gz /var/lib/mysql/*
```

#打包备份的目录及目录下文件

```
mysql50 ~]# scp mysql.tar.gz root@192.168.4.51:/root
```

#传输到新的主机

```
mysql51 ~]# ls
```

```
mysql51 ~]# systemctl stop mysqld #关闭 mysqld 服务
```

```
mysql51 ~]# tar -zxvf /root/mysql.tar.gz -C /var/lib/mysql/
```

#解压到/var/lib/mysql/目录下

```
mysql51 ~]# chown -R mysql:mysql /var/lib/mysql
```

#修改解压后目录及目录下文件的权限

```
mysql51 ~]# ls -ld /var/lib/mysql
```

```
mysql51 ~]# systemctl restart mysqld #重启服务
```

```
mysql51 ~]# ps -C mysqld #查看进程
```

PID	TTY	TIME	CMD
-----	-----	------	-----

23648	?	00:00:00	mysqld #查看端口
-------	---	----------	--------------

```
mysql51 ~]# ss -antulp | grep mysqld
```

```
tcp LISTEN 0 80 :::3306 :::* users:(("mysqld",pid=23648,fd=35))
```

```
mysql51 ~]# mysql -uroot -p123456 #登录并查看数据库及表
```

## 3.2 逻辑备份

**备份命令:mysqldump**

**恢复命令:mysql**

### 3.2.1 策略

**完全备份:**备份所有数据

```
mysqldump -uroot -p 密码 库名 > /路径/xxx.sql
```

**完全恢复:**

```
mysql -uroot -p 密码 [库名] < /路径/xxx.sql
```

**库名表示方式:**

**--all-databases 或 -A**            所有库

**数据库名**                            单个库

**数据库名 表名**                      单张表 #注意为空格,不是.号

**-B 数据库1 数据库2**                多个库 #注意空格和-B

**无论备份还是恢复,都要验证用户权限!!!**

恢复时单个库或单个表需要写库名,恢复单个库需要先创建个同名空库

```
mysql50 ~]# mysqldump -uroot -p123abc -A > /dbbak/all.sql
```

```
mysql50 ~]# mysqldump -uroot -p123abc deb > /dbbak/deb.sql
```

```
mysql50 ~]# mysqldump -uroot -p123abc deb a > /dbbak/deb.a.sql
```

```
mysql50 ~]# mysqldump -uroot -p123abc -B deb mysql > /dbbak/  
debandmysql.sql
```

```
mysql51 ~]# mysql -uroot -p123abc deb < /opt/deb.sql
```

```
mysql51 ~]# mysql -uroot -p123456 deb < /opt/deb.a.sql
```

```
mysql51 ~]# mysql -uroot -p123456 < /opt/debandmysql.sql
```

```
mysql51 ~]# mysql -uroot -p123456 < /opt/all.sql
```

恢复时,若库存在,则覆盖恢复;若不存在,先在 `mysql` 里创建库,再恢复

缺点: 恢复时会锁表,影响其他进程对数据库的读写;新产生的数据不会在备份文件中

**增量备份:**备份上次备份后所有新产生的数据

**差异备份:**备份完全备份后所有新产生的数据

## 四 binlog 日志

### 4.1 binlog 日志

也称为二进制日志

`mysql` 服务日志文件的一种

记录除查询之外的所有 `sql` 命令

可用于数据备份和恢复

配置 `mysql` 主从同步的必要条件

`xxx.index` 索引文件   `xxx.000001` 第 1 个日志   `xxx.00000n` 第 n 个日志

### 4.2 启用日志

`server_id=数字`

**#指定 ID 值(1~255)**

`log_bin=[目录名/文件名开头]`

**#启用 binlog 日志及日志文件名开头**

**max\_binlog\_size=数值 m**            **#指定日志文件容量,默认 1G**

#### 4.2.1 修改配置文件

```
mysql50 ~]# vim /etc/my.cnf
```

```
[mysqld]
```

**log\_bin=/mylog/abc**    **#启用 binlog 日志,并修改存储路径及指定前缀**

**server\_id=50**    **#指定 ID 值**

#### 4.2.2 创建目录/mylog 及修改所属

```
mysql50 ~]# mkdir mylog
```

```
mysql50 ~]# chown mysql:mysql mylog
```

#### 4.2.3 重启服务并查看日志

```
mysql50 ~]# systemctl restart mysqld
```

```
mysql50 ~]# ls /mylog/abc.*
```

#### 4.2.4 登录 mysql 并查看正在使用的日志

```
mysql50 ~]# mysql -uroot -p123456
```

```
mysql> show master status;    #查看正在使用的日志文件
```

#### 4.2.5 查看日志内容

**mysqlbinlog /路径/日志名**

```
mysql50 ~]# mysqlbinlog /mylog/abc.000001
```

### 4.3 手动创建新的日志文件的 4 种方法

#### 4.3.1 systemctl restart mysqld

4.3.2 `mysql> flush logs;`

4.3.3 `mysql50 ~]# mysql -uroot -p123456 -e "flush logs"`

4.3.4 `mysql50 ~]# mysqldump --flush-logs`

## 4.4 日志文件的清理

4.4.1 系统命令行 `rm -rf` 删除所有日志文件及索引

4.4.2 删除指定编号之前的 binlog 日志文件

`mysql> purge master logs to "binlog 文件名";`

4.4.3 删除所有 binlog 日志,重建新日志

`mysql> reset master;`

## 4.5 修改日志记录格式

在 mysql 中,binlog\_format 变量用于指定日志记录格式

日志记录格式:

statement:报表模式      row:行模式      mixed:混合模式

4.5.1 查找变量及查看当前记录格式

`50 mysql> show variables like "%binlog%";`

4.5.2 修改配置文件,将 binlog\_format 设置为 mixed

`mysql50 ~]# vim /etc/my.cnf`

**[mysqld]**

**binlog\_format="mixed"**      #添加行,修改为混合模式

4.5.3 重启服务

```
mysql50 ~]# systemctl restart mysqld
```

#### 4.5.4 在 mysql 内对表插入新值

```
mysql> insert into db3.user(name) values("aaa");
```

```
mysql> insert into db3.user(name) values("aab");
```

```
mysql> insert into db3.user(name) values("aac");
```

#### 4.5.5 查看日志

```
mysql50 ~]# mysqlbinlog /var/lib/log/abc.000002 | grep -i insert
```

```
SET INSERT_ID=26/*!*/;
```

```
insert into db3.user(name) values("aaa")
```

```
SET INSERT_ID=27/*!*/;
```

```
insert into db3.user(name) values("aab")
```

```
SET INSERT_ID=28/*!*/;
```

```
insert into db3.user(name) values("aac")
```

#### 4.6 日志文件如何区分记录多条 sql 命令(日志记录 sql 命令的方式)

```
# at 1056
```

```
#190903 10:20:52 server id 50  end_log_pos 1087 CRC32 0x7cdc4405
```

```
    Xid = 7
```

```
COMMIT/*!*/;
```

at 后的数字为偏移量

#190903 10:20:52 为时间点

4.6.1 偏移量 **--start-position=数字 --stop-positon=数字**

4.6.2 时间点 **--start-datetime="yyyy-mm-dd hh:mm:ss"**  
**--stop-datetime="yyyy-mm-dd hh:mm:ss"**

#### 4.7 读取日志指定范围的 sql 命令恢复数据

```
mysql50 mylog]# scp abc.000002 root@192.168.4.51:/mylog
```

```
mysql51 mylog]# mysqlbinlog --start-position=325
```

```
--stop-position=745 /mylog/abc.000002 | mysql -uroot -p123456
```

```
51 mysql> select * from db3.user; #查看数据是否导入
```