1 用户授权

1.1 什么是用户授权：数据库管理连接服务后添加新用户 设置访问权限 和密码

1.2 命令格式

mysql> grant 权限列表 on 数据库名 to 用户名@"客户端地址" identified by "密码" [ with grant option ] ;

1.3 授权配置项的表示方式：

权限列表 all usage insert,select update(字段名列表)

数据库名 \*.\* 库名.\* 库名.表名

客户端地址 % 192.168.4.51 192.168.4.% localhost

1.4 客户端使用授权用户连接数据库服务器

]# mysql -h数据库服务器ip -u授权用户 -p密码

]# mysql -h192.168.4.50 -uyaya -p123456

例子

50 添加用户

mysql> grant all on \*.\* to mydba@"%" identified by "123qqq...A" with grant option;

mysql> grant select on db3.user to admin@"192.168.4.%" identified by "123qqq...A";

mysql> grant select , insert , update , delete on db3.\* to admin2@"localhost" identified by "123qqq...A" ;

mysql> grant all on \*.\* to root@"192.168.4.51" identified by "123qqq...A" with grant option;

51测试用户

[root@host51 ~]# mysql -h192.168.4.50 -umydba -p123qqq...A

mysql> 对数据做管理的命令都可以执行

[root@host51 ~]# mysql -h192.168.4.50 -uadmin -p123qqq...A

mysql> 对数据做管理

[root@host51 ~]# mysql -h192.168.4.50 -uadmin2 -p123qqq...A

mysql: [Warning] Using a password on the command line interface can be insecure.

ERROR 1045 (28000): Access denied for user 'admin2'@'192.168.4.51' (using password: YES)

[root@host51 ~]#

[root@host50 ~]# mysql -hlocalhost -uadmin2 -p123qqq...A

mysql> 对数据做管理时执行 select insert update delete 之外的sql命令会报错

[root@host51 ~]# mysql -h192.168.4.50 -uroot -p123qqq...A

mysql> grant all on db1.\* to admin3@"%" identified by "123qqq...A";

mysql> grant select on \*.\* to yaya88@"localhost" identified by "123qqq...A";

相关命令

mysql> select @@hostname; 查看主机名

mysql> select host , user from mysql.user;怎么看已有授权用户

show grants for admin3@"%" ; 数据库管理员 怎么看已有授权用户的权限？

连接数据库服务器的用户怎么查看自己的访问权限 show grants;

数据管理员已有授权用户的连接密码

mysql> select password("654321")；

mysql> set password for admin3@"%"=password("654321");

[root@host51 ~]# mysql -h192.168.4.50 -uadmin3 -p654321

mysql> select user();

mysql> show grants;

mysql> set password=password("123qqq...A"); 客户端可以修改连接密码？

MySQL> exit;

[root@host51 ~]# mysql -h192.168.4.50 -uadmin3 -p123qqq...A

[root@host50 ~]# mysql -uroot -ptarena

mysql> drop user admin3@"localhost"; 删除用户

授权库mysql （存储的时grant命令的执行结果）

user表 存储用户名 、客户端地址 全局权限 ( all \*.\* )

desc mysql.user;

select host , user from mysql.user;

select \* from mysql.user \G

show grants for admin3@"%";

db表 存储授权用户对服务器上已有 库的访问

desc mysql.db;

select host , user , db from mysql.db;

mysql> select \* from mysql.db where db="db1" \G

mysql> show grants for admin3@"%";

mysql> update mysql.db set Delete\_priv="N" where user="admin3";

mysql> flush privileges;

tables\_priv表 存储授权用户对服务器上已有 表的访问

MySQL> desc mysql.tables\_priv;

mysql> select host , user , db , table\_name from mysql.tables\_priv;

mysql> select \* from mysql.tables\_priv where Table\_name="user" \G

mysql> update mysql.tables\_priv set Table\_priv="insert,delete,select" where Table\_name="user";

mysql> flush privileges;

show grants for admin@"192.168.4.%" ;

columns\_priv表 存储授权用户对表里 字段的访问权限

desc mysql.columns\_priv;

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

Empty set (0.00 sec)

mysql> grant select , update(name,password) on db3.user to admin4@"%" identified by "123qqq...A";

Query OK, 0 rows affected, 1 warning (0.00 sec)

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

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

| Host | Db | User | Table\_name | Column\_name | Timestamp | Column\_priv |

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

| % | db3 | admin4 | user | name | 0000-00-00 00:00:00 | Update |

| % | db3 | admin4 | user | password | 0000-00-00 00:00:00 | Update |

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

2 rows in set (0.01 sec)

mysql>

目的：查看表记录可以获取用户权限；也可以通过更新记录，修改用户权限

撤销权限(删除已有用户的访问权限）

MySQL> revoke 权限列表 on 库名.表 from 用户名@"客户端地址";

查看授权用户权限

mysql> select host , user from mysql.user;

mysql> show grants for mydba@"%";

mysql> revoke grant option on \*.\* from mydba@"%";

mysql> revoke delete , drop on \*.\* from mydba@"%";

mysql> revoke all on \*.\* from mydba@"%";

mysql> show grants for mydba@"%";

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

| Grants for mydba@% |

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

| GRANT USAGE ON \*.\* TO 'mydba'@'%' |

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

1 row in set (0.00 sec)

mysql>

[root@host51 ~]# mysql -h192.168.4.50 -uyaya99 -p123456

mysql> show grants;

root密码管理（操作系统的管理员root用户）

修改数据库管理员root用户本机登录密码

命令格式[root@host50 ~]# mysqladmin -uroot -p旧密码 password "新密码"

例子

[root@host50 ~]# mysqladmin -uroot -ptarena password "666888"

[root@host50 ~]# mysql -uroot -p666888 #使用新密码登录

恢复数据库管理员root用户本机登录密码（忘记密码）

[root@host50 ~]# systemctl stop mysqld

[root@host50 ~]# vim /etc/my.cnf

[mysqld]

skip-grant-tables

#validate\_password\_policy=0

#validate\_password\_length=6

:wq

[root@host50 ~]# systemctl start mysqld

[root@host50 ~]# mysql

mysql> select user , host , authentication\_string from mysql.user;

mysql> update mysql.user set authentication\_string=password("123qqq...A")

where user="root" and host="localhost" ;

mysql> flush privileges;

mysql> exit

[root@host50 ~]# vim /etc/my.cnf

[mysqld]

#skip-grant-tables

validate\_password\_policy=0

validate\_password\_length=6

:wq

[root@host50 ~]# systemctl restart mysqld

[root@host50 ~]# mysql -uroot -p123qqq...A

数据备份的相关概念？

备份？ 备份的是数据 备份的目的：恢复数据

备份方式：

物理备份 把数据库目录 拷贝 一份 存放到其他的 存储设备上

物理恢复 把备份的文件拷贝回数据库目录，具体步骤如下：

停止数据库服务

清空数据库目录下数据

拷贝备份目录下的文件 到 数据库目录下

递归修改数据库目录的所有者和用户组 为mysql用户

启动数据库服务

管理员登录查看数据

演示

[root@host50 ~]# cp -r /var/lib/mysql /mysql.bak

[root@host50 ~]# scp -r /mysql.bak root@192.168.4.51:/opt/

[root@host51 ~]# systemctl stop mysqld

[root@host51 ~]# rm -rf /var/lib/mysql/\*

[root@host51 ~]#

[root@host51 ~]# ls /var/lib/mysql

[root@host51 ~]#

[root@host51 ~]# ls /opt/mysql.bak/

[root@host51 ~]# cp -r /opt/mysql.bak/\* /var/lib/mysql/

[root@host51 ~]# chown -R mysql:mysql /var/lib/mysql

[root@host51 ~]# systemctl start mysqld

[root@host51 ~]# mysql -uroot -p密码

逻辑备份：使用mysql软件提供备份命令做备份，在执行备份命令时，根据已有的库表记录生成对应的sql命令 保存定义的文件里。

数据备份策略：

完全备份：备份所有数据

只 备份 1张表的所有数据

只 备份 1个库的所有数据

备份1台数据库服务器的所有数据

只备份新产生的数据：

差异备份：备份完全备份后，所有新产生的数据

增量备份：备份上次备份后，所有新产生的数据

工作中的备份策略

完全备份+差异备份 （用当前的所有数据减去第1天的数据 剩下的就是 要备份的新数据）

完全备份+增量备份 （判断新数据，只和前一天比较）

休息+订餐 到 11：05

2 完全备份 （数据备份与恢复）

数据完全备份命令 mysqldump

]# mysqldump -uroot -p密码 数据库名 > 文件名.sql

数据库名的表示方式？

--all-databases 或 -A //所有库

数据库名 //单个库

数据库名 表名 //单张表

-B 数据库1 数据库2 //多个库

例子：

[root@host50 ~]# mkdir /bakdir

[root@host50 ~]# mysqldump -uroot -p123qqq...A --all-databases > /bakdir/allbak.sql

[root@host50 ~]# mysqldump -uroot -p123qqq...A db1 > /bakdir/db1.sql

[root@host50 ~]# mysqldump -uroot -p123qqq...A db3 user > /bakdir/db3\_user.sql

[root@host50 ~]# mysqldump -uroot -p123qqq...A -B db1 db3 > /bakdir/twodb.sql

[root@host50 ~]# ls /bakdir/

allbak.sql db1.sql db3\_user.sql twodb.sql

[root@host50 ~]#

数据完全恢复命令 mysql

]# mysql -uroot -p密码 库名 < 文件名.sql

例子

mysql> create database db3;

mysql>exit

[root@host51 ~]# mysql -uroot -p123qqq...A db3 < /root/db3\_user.sql

[root@host51 ~]# mysql -uroot -p123qqq...A -e ' select \* from db3.user '

[root@host51 ~]# mysql -uroot -p123qqq...A -e ' select count(\*) from db3.user '

删除已有库 只有不重启数据库服务器 就可以使用备份文件恢复数据

[root@host51 ~]# mysql -uroot -p123qqq...A < /root/allbak.sql

3 数据的增量备份与恢复

3.1 启用binlog日志实现数据的时时备份

3.1.1 binlog日志介绍 ？

3.1.2 启用binlog日志

]# vim /etc/my.cnf

[mysqld]

server\_id=50

log\_bin

:wq

]# systemctl restart mysqld

]# mysql -uroot -p密码

mysql> show master status;

mysql> system ls /var/lib/mysql/

mysql> insert into db3.user(name,uid) values("bob",3);

mysql> show master status;

mysql> select \* from db3.user;

mysql> show master status;

修改日志文件存放的目录和文件名

[root@host50 ~]# mkdir /mylog

[root@host50 ~]# chown mysql /mylog

[root@host50 ~]# ls -ld /mylog

drwxr-xr-x. 2 mysql root 6 8月 18 14:44 /mylog

[root@host50 ~]#

[root@host50 ~]# setenforce 0

]# vim /etc/my.cnf

[mysqld]

server\_id=50

#log\_bin

log\_bin=/mylog/plj

:wq

[root@host50 ~]# systemctl restart mysqld

[root@host50 ~]# ls /mylog

plj.000001 plj.index

[root@host50 ~]#

mysql: [Warning] Using a password on the command line interface can be insecure.

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

| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |

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

| plj.000001 | 154 | | | |

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

[root@host50 ~]#

[root@host50 ~]# mysql -uroot -p123qqq...A -e 'insert into db3.user(name) values("aaa")'

mysql: [Warning] Using a password on the command line interface can be insecure.

[root@host50 ~]#

[root@host50 ~]# mysql -uroot -p123qqq...A -e 'insert into db3.user(name) values("bbb")'

mysql: [Warning] Using a password on the command line interface can be insecure.

[root@host50 ~]#

[root@host50 ~]# mysql -uroot -p123qqq...A -e 'show master status'

mysql: [Warning] Using a password on the command line interface can be insecure.

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

| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |

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

| plj.000001 | 696 | | | |

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

[root@host50 ~]#

3.1.3 查看日志文件内容

]# mysqlbinlog 目录名/日志文件名

[root@host50 ~]# mysqlbinlog /mylog/plj.000001

3.1.4 手动创建新的日志文件

]# systemctl restart mysqld

[root@host50 ~]# mysql -uroot -p123qqq...A -e 'flush logs'

mysql> flush logs;

[root@host50 ~]# mysqldump -uroot -p123qqq...A --flush-logs db3 user > /opt/user2.sql

3.1.5 删除已有的日志文件

mysql> purge master logs to "plj.000004"; 删除指定编号之前的binlog日志文件

mysql> reset master ; 重建新日志

]# rm -rf \*.0000\* \*.index 使用系统命令2种全删除

3.2 使用日志恢复数据

使用日志恢复所有数据，命令格式

]#mysqlbinlog /目录名/日志名 | mysql -uroot -p密码

[root@host50 ~]#mysql> select count(\*) from db3.user;

[root@host50 ~]# mysqldump -uroot -p123qqq...A --flush-logs db3 user > /root/user.sql

mysqldump: [Warning] Using a password on the command line interface can be insecure.

[root@host50 ~]# scp /root/user.sql root@192.168.4.51:/root/

root@192.168.4.51's password:

user.sql 100% 3543 184.2KB/s 00:00

[root@host50 ~]#

[root@host50 ~]# mysql -uroot -p123qqq...A -e ' show master status'

mysql: [Warning] Using a password on the command line interface can be insecure.

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

| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |

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

| plj.000002 | 154 | | | |

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

[root@host50 ~]#

[root@host50 ~]# mysql -uroot -p123qqq...A -e 'insert into db3.user(name,uid) values("a1",33)'

mysql: [Warning] Using a password on the command line interface can be insecure.

[root@host50 ~]# mysql -uroot -p123qqq...A -e 'insert into db3.user(name,uid) values("a2",33)'

mysql: [Warning] Using a password on the command line interface can be insecure.

[root@host50 ~]# mysql -uroot -p123qqq...A -e 'insert into db3.user(name,uid) values("a3",33)'

mysql: [Warning] Using a password on the command line interface can be insecure.

[root@host50 ~]# mysql -uroot -p123qqq...A -e 'insert into db3.user(name,uid) values("a4",33)'

mysql: [Warning] Using a password on the command line interface can be insecure.

[root@host50 ~]#

[root@host50 ~]# mysql -uroot -p123qqq...A -e ' show master status'

mysql: [Warning] Using a password on the command line interface can be insecure.

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

| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |

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

| plj.000002 | 1250 | | | |

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

[root@host50 ~]# scp /mylog/plj.000002 root@192.168.4.51:/opt/

[root@host51 ~]# mysql -uroot -p123qqq...A db3 < /root/user.sql

mysql: [Warning] Using a password on the command line interface can be insecure.

[root@host51 ~]#

[root@host51 ~]# mysql -uroot -p123qqq...A -e 'select count(\*) from db3.user'

mysql: [Warning] Using a password on the command line interface can be insecure.

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

| count(\*) |

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

| 29 |

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

[root@host51 ~]# ls /opt/plj.000002

/opt/plj.000002

[root@host51 ~]# mysqlbinlog /opt/plj.000002 | mysql -uroot -p123qqq...A

mysql: [Warning] Using a password on the command line interface can be insecure.

[root@host51 ~]#

[root@host51 ~]# mysql -uroot -p123qqq...A -e 'select count(\*) from db3.user'

mysql: [Warning] Using a password on the command line interface can be insecure.

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

| count(\*) |

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

| 33 |

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

[root@host51 ~]#

使用日志恢复部分数据 , 命令格式

日志记录sql命令的方式?

记录命令执行的时间

记录命令执行的偏移量

]# mysqlbinlog 选项 /目录名/文件名 | mysql -uroot -p密码

读偏移量范围内记录的sql命令

--start-position=数字 --stop-position=数字

都时间范围内记录的sql命令

--start-datetime="yyyy/mm/dd hh:mm:ss" --stop-datetime="yyyy/mm/dd hh:mm:ss"

修改日志文件的记录格式查看

[root@host50 mylog]# mysql -uroot -p123qqq...A -e 'show variables like "binlog\_format"'

mysql: [Warning] Using a password on the command line interface can be insecure.

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

| Variable\_name | Value |

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

| binlog\_format | ROW |

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

[root@host50 mylog]#

修改日志格式

]# vim /etc/my.cnf

[mysqld]

binlog\_format="mixed"

:wq

[root@host50 mylog]# systemctl restart mysqld

[root@host50 mylog]# mysql -uroot -p123qqq...A -e 'show variables like "binlog\_format"'

mysql: [Warning] Using a password on the command line interface can be insecure.

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

| Variable\_name | Value |

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

| binlog\_format | MIXED |

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

[root@host50 mylog]# mysql -uroot -p123qqq...A -e 'show master status'

mysql: [Warning] Using a password on the command line interface can be insecure.

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

| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |

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

| plj.000003 | 154 | | | |

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

[root@host50 mylog]#

[root@host50 mylog]# mysql -uroot -p123qqq...A -e 'insert into db3.user(name,uid) values("c1",5)'

[root@host50 mylog]# mysql -uroot -p123qqq...A -e 'insert into db3.user(name,uid) values("c2",5)'

[root@host50 mylog]# mysql -uroot -p123qqq...A -e 'insert into db3.user(name,uid) values("c3",5)'

[root@host50 mylog]# mysql -uroot -p123qqq...A -e "delete from db3.user where name like 'a%'"

[root@host50 mylog]# mysql -uroot -p123qqq...A -e 'show master status'

mysql: [Warning] Using a password on the command line interface can be insecure.

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

| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |

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

| plj.000003 | 1382 | | | |

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

[root@host50 mylog]#

[root@host50 mylog]# mysqlbinlog /mylog/plj.000003 | grep insert

insert into db3.user(name,uid) values("c1",5)

insert into db3.user(name,uid) values("c2",5)

insert into db3.user(name,uid) values("c3",5)

[root@host50 mylog]#

[root@host50 mylog]# mysqlbinlog /mylog/plj.000003 | grep delete

delete from db3.user where name like 'a%'

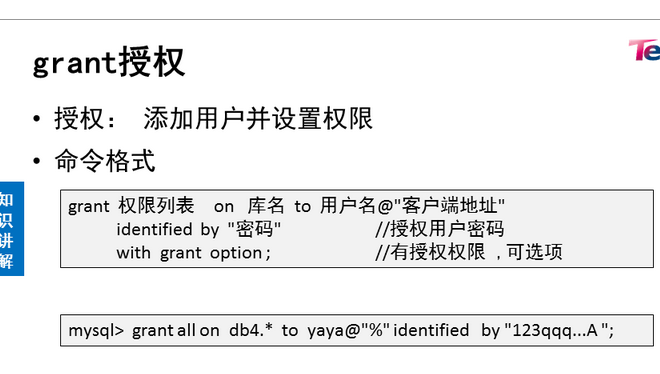
[root@host50 mylog]#

[root@host50 mylog]#scp /mylog/plj.000003 root@192.168.4.51:/root/

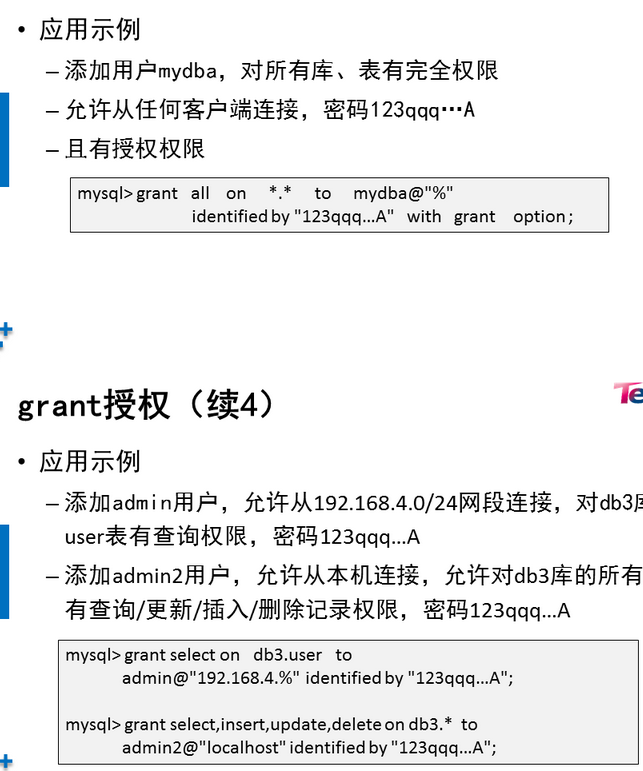
[root@host51 ~]# mysqlbinlog --start-position=325 --stop-position=470 /root/plj.000003 | mysql -uroot -p123qqq...A

[root@host51 ~]# mysqlbinlog --start-datetime="2020/08/18 16:56:36" --stop-datetime="2020/08/18 16:59:29" /root/plj.000003 | mysql -uroot -p123qqq...A

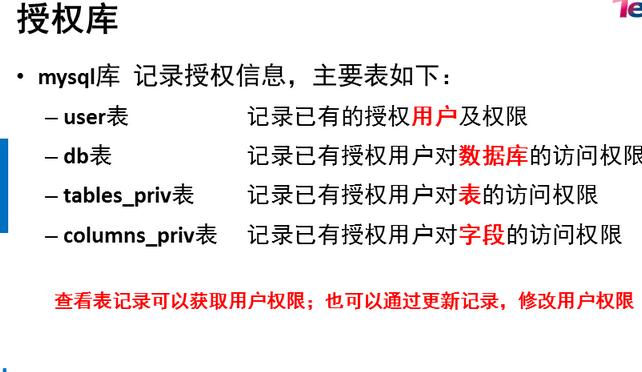
[root@host51 ~]# mysql -uroot -p123qqq...A -e "select \* from db3.user WHERE name like 'c%'"



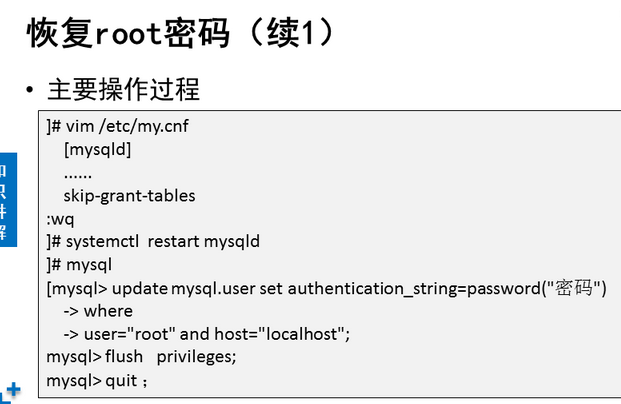




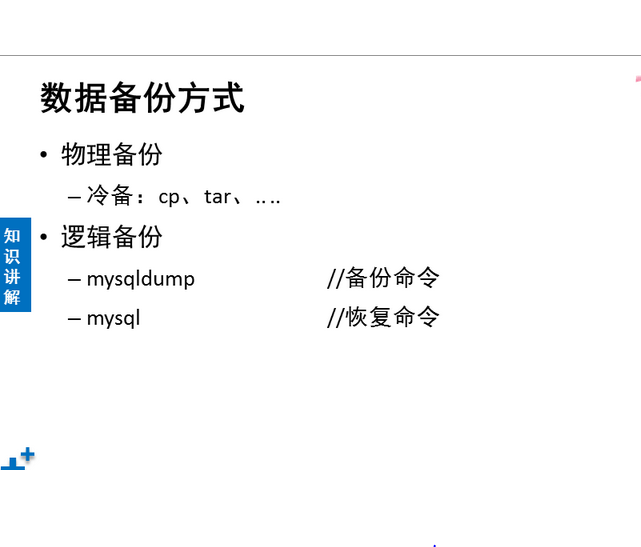




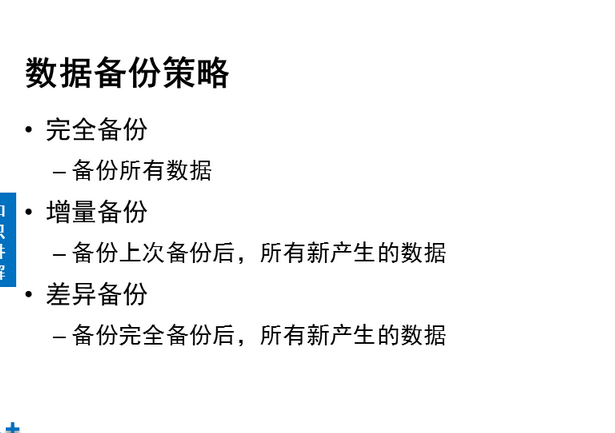


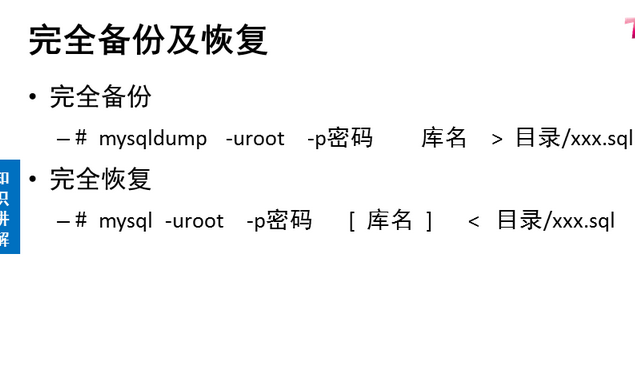


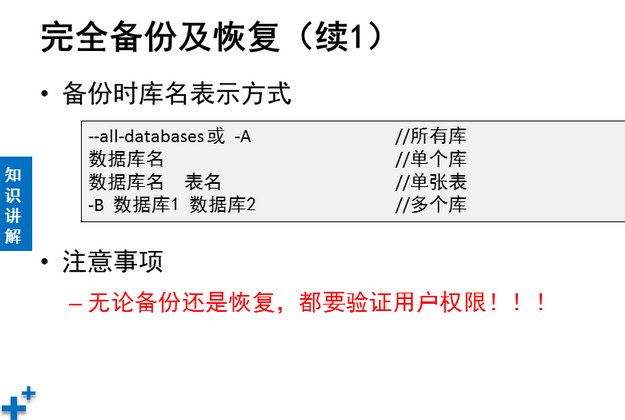


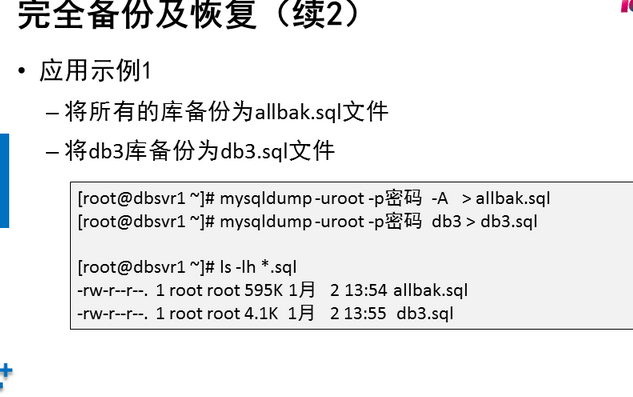




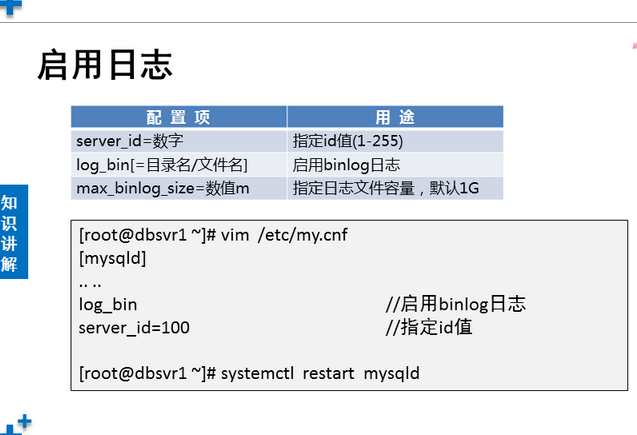






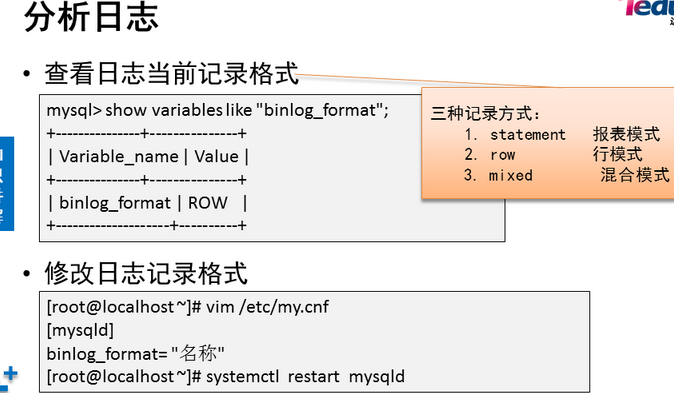


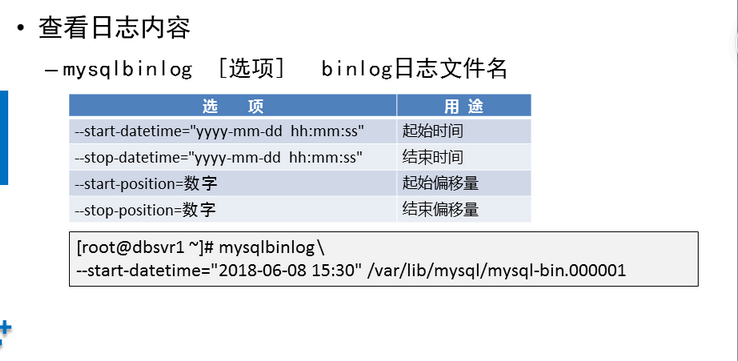














**NSD RDBM1 DAY04**

1. [案例1： 用户授权](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/RDBMS1/DAY04/CASE/01/index.html#case1)
2. [案例2：root密码](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/RDBMS1/DAY04/CASE/01/index.html#case2)
3. [案例3：数据备份与恢复](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/RDBMS1/DAY04/CASE/01/index.html#case3)
4. [案例4：binlog日志](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/RDBMS1/DAY04/CASE/01/index.html#case4)
5. [案例5：使用binlog日志恢复数据](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/RDBMS1/DAY04/CASE/01/index.html#case5)

**1 案例1： 用户授权**

**1.1 问题**

* 允许192.168.4.0/24网段主机使用root连接数据库服务器，对所有库和所有表有完全权限、密码为123qqq…A 。
* 添加用户dba007，对所有库和所有表有完全权限、且有授权权限，密码为123qqq…A 客户端为网络中的所有主机。
* 撤销root从本机访问权限，然后恢复。
* 允许任意主机使用webuser用户连接数据库服务器，仅对webdb库有完全权限，密码为123qqq…A 。
* 撤销webuser的权限，使其仅有查询记录权限。

**1.2 步骤**

实现此案例需要按照如下步骤进行。

步骤一：用户授权

1）允许root从192.168.4.0/24访问，对所有库表有完全权限，密码为123qqq…A

授权之前，从192.168.4.0/24网段的客户机访问时，将会被拒绝：

1. [root@host120 ~]# mysql -u root -p -h 192.168.4.10
2. Enter password:                                 //输入正确的密码
3. ERROR 2003 (HY000): Host '192.168.4.120' is not allowed to connect to this MySQL server

授权操作，此处可设置与从localhost访问时不同的密码：

1. mysql> GRANT all ON \*.\* TO root@'192.168.4.%' IDENTIFIED BY 'tarena';
2. Query OK, 0 rows affected (0.00 sec)

再次从192.168.4.0/24网段的客户机访问时，输入正确的密码后可登入：

1. [root@host120 ~]# mysql -u root -p -h 192.168.4.10
2. Enter password:
3. Welcome to the MySQL monitor. Commands end with ; or \g.
4. Your MySQL connection id is 20
5. Server version: 5.7.17 MySQL Community Server (GPL)
6. Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.
7. Oracle is a registered trademark of Oracle Corporation and/or its
8. affiliates. Other names may be trademarks of their respective
9. owners.
10. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
11. mysql>

从网络登入后，测试新建一个库、查看所有库：

1. mysql> CREATE DATABASE rootdb;                 //创建新库rootdb
2. Query OK, 1 row affected (0.06 sec)
3. mysql> SHOW DATABASES;
4. +--------------------+
5. | Database |
6. +--------------------+
7. | information\_schema |
8. | home |
9. | mysql |
10. | performance\_schema |
11. | rootdb | //新建的rootdb库
12. | sys |
13. | userdb |
14. +--------------------+
15. 7 rows in set (0.01 sec)

2）在Mysql服务器上建立一个管理账号dba007，对所有库完全控制，并赋予其授权的权限新建账号并授权：

1. mysql> GRANT all ON \*.\* TO dba007@localhost
2. -> IDENTIFIED BY '123qqq…A '
3. -> WITH GRANT OPTION;
4. Query OK, 0 rows affected (0.00 sec)

查看dba007的权限：

1. mysql> SHOW GRANTS FOR dba007@localhost;
2. +-----------------------------------------------------------------------+
3. | Grants for dba007@localhost |
4. +-----------------------------------------------------------------------+
5. | GRANT ALL PRIVILEGES ON \*.\* TO 'dba007'@'localhost' WITH GRANT OPTION |
6. +-----------------------------------------------------------------------+
7. 1 row in set (0.00 sec)

3）撤销root从本机访问的权限，然后恢复

注意：如果没有事先建立其他管理账号，请不要轻易撤销root用户的本地访问权限，否则恢复起来会比较困难，甚至不得不重装数据库。

撤销root对数据库的操作权限：

1. mysql> REVOKE all ON \*.\* FROM root@localhost;
2. Query OK, 0 rows affected (0.00 sec)
3. mysql> SHOW GRANTS FOR root@localhost;
4. +--------------------------------------------------------------+
5. | Grants for root@localhost |
6. +--------------------------------------------------------------+
7. | GRANT USAGE ON \*.\* TO 'root'@'localhost' WITH GRANT OPTION |
8. | GRANT PROXY ON ''@'' TO 'root'@'localhost' WITH GRANT OPTION |
9. +--------------------------------------------------------------+
10. 2 rows in set (0.00 sec)

验证撤销后的权限效果：

1. mysql> exit                                     //退出当前MySQL连接
2. Bye
3. [root@dbsvr1 ~]# mysql -u root -p                 //重新以root从本地登入
4. Enter password:
5. Welcome to the MySQL monitor. Commands end with ; or \g.
6. Your MySQL connection id is 6
7. Server version: 5.6.15 MySQL Community Server (GPL)
8. Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.
9. Oracle is a registered trademark of Oracle Corporation and/or its
10. affiliates. Other names may be trademarks of their respective
11. owners.
12. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
13. mysql> CREATE DATABASE newdb2014;                 //尝试新建库失败
14. ERROR 1044 (42000): Access denied for user 'root'@'localhost' to database 'newdb2014'
15. mysql> DROP DATABASE rootdb;                         //尝试删除库失败
16. ERROR 1044 (42000): Access denied for user 'root'@'localhost' to database 'rootdb'

尝试以当前的root用户恢复权限，也会失败（无权更新授权表）：

1. mysql> GRANT all ON \*.\* TO root@localhost IDENTIFIED BY '1234567';
2. ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)

怎么办呢？

退出当前MySQL连接，以上一步添加的管理账号dba007登入：

1. mysql> exit                                         //退出当前MySQL连接
2. Bye
3. [root@dbsvr1 ~]# mysql -u dba007 -p                 //以另一个管理账号登入
4. Enter password:
5. Welcome to the MySQL monitor. Commands end with ; or \g.
6. Your MySQL connection id is 24
7. Server version: 5.7.17 MySQL Community Server (GPL)
8. Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.
9. Oracle is a registered trademark of Oracle Corporation and/or its
10. affiliates. Other names may be trademarks of their respective
11. owners.
12. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

由管理账号dba007重新为root添加本地访问权限：

1. mysql> GRANT all ON \*.\* TO root@localhost IDENTIFIED BY '1234567';
2. Query OK, 0 rows affected (0.00 sec)
3. mysql> SHOW GRANTS FOR root@localhost;             //查看恢复结果
4. +---------------------------------------------------------------------+
5. | Grants for root@localhost |
6. +---------------------------------------------------------------------+
7. | GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'localhost' WITH GRANT OPTION |
8. | GRANT PROXY ON ''@'' TO 'root'@'localhost' WITH GRANT OPTION |
9. +---------------------------------------------------------------------+
10. 2 rows in set (0.00 sec)

退出，再重新以root登入，测试一下看看，权限又恢复了吧：

1. mysql> exit                                     //退出当前MySQL连接
2. Bye
3. [root@dbsvr1 ~]# mysql -u root -p                 //重新以root登入
4. Enter password:
5. Welcome to the MySQL monitor. Commands end with ; or \g.
6. Your MySQL connection id is 25
7. Server version: 5.7.17 MySQL Community Server (GPL)
8. Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.
9. Oracle is a registered trademark of Oracle Corporation and/or its
10. affiliates. Other names may be trademarks of their respective
11. owners.
12. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
13. mysql> CREATE DATABASE newdb2014;                 //成功创建新库
14. Query OK, 1 row affected (0.00 sec)

4）允许webuser从任意客户机登录，只对webdb库有完全权限，密码为 123qqq…A

添加授权：

1. mysql> GRANT all ON webdb.\* TO webuser@'%' IDENTIFIED BY '888888';
2. Query OK, 0 rows affected (0.00 sec)

查看授权结果：

1. mysql> SHOW GRANTS FOR webuser@'%';
2. +----------------------------------------------------+
3. | Grants for webuser@% |
4. +----------------------------------------------------+
5. | GRANT USAGE ON \*.\* TO 'webuser'@'%' |
6. | GRANT ALL PRIVILEGES ON `webdb`.\* TO 'webuser'@'%' |
7. +----------------------------------------------------+
8. 2 rows in set (0.00 sec)

5）撤销webuser的完全权限，改为查询权限

撤销所有权限：

1. mysql> REVOKE all ON webdb.\* FROM webuser@'%';
2. Query OK, 0 rows affected (0.00 sec)

只赋予查询权限：

1. mysql> GRANT select ON webdb.\* TO webuser@'%';
2. Query OK, 0 rows affected (0.00 sec)

确认授权更改结果：

1. mysql> SHOW GRANTS FOR webuser@'%';
2. +--------------------------------------------+
3. | Grants for webuser@% |
4. +--------------------------------------------+
5. | GRANT USAGE ON \*.\* TO 'webuser'@'%' |
6. | GRANT SELECT ON `webdb`.\* TO 'webuser'@'%' |
7. +--------------------------------------------+
8. 2 rows in set (0.00 sec)

**2 案例2：root密码**

**2.1 问题**

具体要求如下：

* 恢复管理员root密码 123qqq…A
* 重置管理员root密码 A…qqq321

**2.2 步骤**

实现此案例需要按照如下步骤进行。

步骤一：恢复管理员root密码

1）首先停止已运行的MySQL服务程序

1. [root@dbsvr1 ~]# systemctl stop mysqld.service         //停止服务
2. [root@dbsvr1 ~]# systemctl status mysqld.service      //确认状态
3. mysqld.service - MySQL Server
4. Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled)
5. Active: inactive (dead) since 五 2017-04-07 23:01:38 CST; 21s ago
6. Docs: man:mysqld(8)
7. http://dev.mysql.com/doc/refman/en/using-systemd.html
8. Process: 20260 ExecStart=/usr/sbin/mysqld --daemonize --pid-file=/var/run/mysqld/mysqld.pid $MYSQLD\_OPTS (code=exited, status=0/SUCCESS)
9. Process: 20238 ExecStartPre=/usr/bin/mysqld\_pre\_systemd (code=exited, status=0/SUCCESS)
10. Main PID: 20262 (code=exited, status=0/SUCCESS)

2）然后跳过授权表启动MySQL服务程序

这一步主要利用mysqld的 --skip-grant-tables选项

修改my.cnf配置，添加 skip\_grant\_tables=1启动设置：

1. [root@dbsvr1 ~]# vim /etc/my.cnf
2. [mysqld]
3. skip\_grant\_tables
4. .. ..
5. [root@dbsvr1 ~]# systemctl start mysqld.service
6. [root@dbsvr1 ~]# service mysql status
7. mysqld.service - MySQL Server
8. Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled)
9. Active: active (running) since 五 2017-04-07 23:40:20 CST; 40s ago
10. Docs: man:mysqld(8)
11. http://dev.mysql.com/doc/refman/en/using-systemd.html
12. Process: 11698 ExecStart=/usr/sbin/mysqld --daemonize --pid-file=/var/run/mysqld/mysqld.pid $MYSQLD\_OPTS (code=exited, status=0/SUCCESS)
13. Process: 11676 ExecStartPre=/usr/bin/mysqld\_pre\_systemd (code=exited, status=0/SUCCESS)
14. Main PID: 11701 (mysqld)
15. CGroup: /system.slice/mysqld.service
16. └─11701 /usr/sbin/mysqld --daemonize --pid-file=/var/run/mysqld/mysqld.p...

3）使用mysql命令连接到MySQL服务，重设root的密码

由于前一步启动的MySQL服务跳过了授权表，所以可以root从本机直接登录

1. [root@dbsvr1 ~]# mysql //直接回车即可
3. Welcome to the MySQL monitor. Commands end with ; or \g.
4. Your MySQL connection id is 4
5. Server version: 5.7.17 MySQL Community Server (GPL)
6. Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.
7. Oracle is a registered trademark of Oracle Corporation and/or its
8. affiliates. Other names may be trademarks of their respective
9. owners.
10. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
11. mysql>

进入 mysql> 环境后，通过修改mysql库中user表的相关记录，重设root用户从本机登录的密码：

1. mysql> UPDATE mysql.user SET authentication\_string=PASSWORD('123qqq…A')
2. -> WHERE user='root' AND host='localhost';             //重设root的密码
3. Query OK, 1 row affected, 1 warning (0.00 sec)
4. Rows matched: 1 Changed: 1 Warnings: 1
5. mysql> FLUSH PRIVILEGES;                                 //刷新授权表
6. Query OK, 0 rows affected (0.01 sec)
7. mysql> exit                                             //退出mysql> 环境
8. Bye

通过执行“FLUSH PRIVILEGES;”可使授权表立即生效，对于正常运行的MySQL服务，也可以用上述方法来修改密码，不用重启服务。本例中因为是恢复密码，最好重启MySQL服务程序，所以上述“FLUSH PRIVILEGES;”操作可跳过。

4）重新以正常方式启动MySQL服务程序，验证新密码

如果前面是修改/etc/my.cnf配置的方法来跳过授权表，则重置root密码后，应去除相应的设置以恢复正常：

1. [root@dbsvr1 ~]# vim /etc/my.cnf
2. [mysqld]
3. #skip\_grant\_tables=1                             //注释掉或删除此行
4. .. ..

按正常方式，通过mysql脚本重启服务即可：

1. [root@dbsvr1 ~]# systemctl restart mysqld.service

验证无密码登录时，将会被拒绝：

1. [root@dbsvr1 ~]# mysql -u root
2. Enter password: //没有跳过授权表回车会报错
3. ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)

只有提供重置后的新密码，才能成功登入：

1. [root@dbsvr1 ~]# mysql -uroot –p123qqq…A
2. Welcome to the MySQL monitor. Commands end with ; or \g.
3. Your MySQL connection id is 4
4. Server version: 5.7.17 MySQL Community Server (GPL)
5. Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.
6. Oracle is a registered trademark of Oracle Corporation and/or its
7. affiliates. Other names may be trademarks of their respective
8. owners.
9. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
10. mysql>

步骤二：重置管理员root密码

正常的前提是：已知当前MySQL管理用户（root）的密码。

1）方法1，在Shell命令行下设置

使用mysqladmin管理工具，需要验证旧的密码。比如，以下操作将会把root的密码设置为 1234567：

1. [root@dbsvr1 ~]# mysqladmin -uroot -p password 'A…qqq321'
2. Enter password: //验证原来的密码
3. mysqladmin: [Warning] Using a password on the command line interface can be insecure.
4. Warning: Since password will be sent to server in plain text, use ssl connection to ensure password safety. //提示明文修改不安全，并不是报错
5. [root@dbsvr1 ~]# mysql -uroot –pA…qqq321 //使用修改后的密码登录
6. Welcome to the MySQL monitor. Commands end with ; or \g.
7. Your MySQL connection id is 4
8. Server version: 5.7.17 MySQL Community Server (GPL)
9. Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.
10. Oracle is a registered trademark of Oracle Corporation and/or its
11. affiliates. Other names may be trademarks of their respective
12. owners.
13. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
14. mysql>

步骤三：修改管理员root密码的其他方法

1）方法1，以root登入mysql> 后，使用SET PASSWORD指令设置

这个与新安装MySQL-server后首次修改密码时要求的方式相同，平时也可以用：

1. mysql> SET PASSWORD FOR root@localhost=PASSWORD('1234567');
2. Query OK, 0 rows affected, 1 warning (0.00 sec)

2）方法2，以root登入mysql> 后，使用GRANT授权工具设置

这个是最常见的用户授权方式（下一节会做更多授权的练习）：

1. mysql> GRANT all ON \*.\* TO root@localhost IDENTIFIED BY '1234567';
2. Query OK, 0 rows affected, 1 warning (0.00 sec)

3）方法3，以root登入mysql> 后，使用UPDATE更新相应的表记录

这种方法与恢复密码时的操作相同：

1. mysql> UPDATE mysql.user SET authentication\_string=PASSWORD('1234567')
2. -> WHERE user='root' AND host='localhost';         //重设root的密码
3. Query OK, 0 rows affected, 1 warning (0.00 sec)
4. Rows matched: 1 Changed: 0 Warnings: 1
5. mysql> FLUSH PRIVILEGES;                                 //刷新授权表
6. Query OK, 0 rows affected (0.00 sec)

在上述方法中，需要特别注意：当MySQL服务程序以 skip-grant-tables 选项启动时，如果未执行“FLUSH PRIVILEGES;”操作，是无法通过SET PASSWORD或者GRANT方式来设置密码的。比如，验证这两种方式时，都会看到ERROR 1290的出错提示：

1. mysql> SET PASSWORD FOR root@localhost=PASSWORD('1234567');
2. ERROR 1290 (HY000): The MySQL server is running with the --skip-grant-tables option so it cannot execute this statement
3. mysql> GRANT all ON \*.\* TO root@localhost IDENTIFIED BY '1234567';
4. ERROR 1290 (HY000): The MySQL server is running with the --skip-grant-tables option so it cannot execute this statement

**3 案例3：数据备份与恢复**

**3.1 问题**

具体要求如下：

* 练习mysqldump命令的使用
* 使用 mysql 命令恢复删除的数据

**3.2 步骤**

实现此案例需要按照如下步骤进行。

步骤一：练习mysqldump命令的使用

1）备份MySQL服务器上的所有库

将所有的库备份为mysql-all.sql文件：

1. [root@dbsvr1 ~]# mysqldump -u root -p --all-databases > /root/alldb.sql
2. Enter password:                                 //验证口令
3. [root@dbsvr1 mysql]# file /root/alldb.sql         //确认备份文件类型
4. /root/alldb.sql: UTF-8 Unicode English text, with very long lines

查看备份文件alldb.sql的部分内容：

1. [root@dbsvr1 ~]# grep -vE '^/|^-|^$' /root/alldb.sql | head -15
2. CREATE DATABASE /\*!32312 IF NOT EXISTS\*/ `home` /\*!40100 DEFAULT CHARACTER SET latin1 \*/;
3. USE `home`;
4. DROP TABLE IF EXISTS `biao01`;
5. CREATE TABLE `biao01` (
6. `id` int(2) NOT NULL,
7. `name` varchar(8) DEFAULT NULL
8. ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
9. LOCK TABLES `biao01` WRITE;
10. UNLOCK TABLES;
11. DROP TABLE IF EXISTS `biao02`;
12. CREATE TABLE `biao02` (
13. `id` int(4) NOT NULL,
14. `name` varchar(8) DEFAULT NULL,
15. PRIMARY KEY (`id`)
16. ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
17. .. ..

注意：若数据库都使用MyISAM存储引擎，可以采用冷备份的方式，直接复制对应的数据库目录即可；恢复时重新复制回来就行。

2）只备份指定的某一个库

将userdb库备份为userdb.sql文件：

1. [root@dbsvr1 ~]# mysqldump -u root -p userdb > userdb.sql
2. Enter password:                                 //验证口令

查看备份文件userdb.sql的部分内容：

1. [root@dbsvr1 ~]# grep -vE '^/|^-|^$' /root/userdb.sql
2. DROP TABLE IF EXISTS `stu\_info`;
3. CREATE TABLE `stu\_info` (
4. `name` varchar(12) NOT NULL,
5. `gender` enum('boy','girl') DEFAULT 'boy',
6. `age` int(3) NOT NULL
7. ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
8. LOCK TABLES `stu\_info` WRITE;
9. .. ..

3）同时备份指定的多个库

同时备份mysql、userdb库，保存为mysql+userdb.sql文件：

1. [root@dbsvr1 ~]# mysqldump -u root -p -B mysql userdb > mysql+test+userdb.sql
2. Enter password:                                 //验证口令

查看备份文件userdb.sql的部分内容：

1. [root@dbsvr1 ~]# grep '^CREATE DATA' /root/mysql+userdb.sql
2. CREATE DATABASE /\*!32312 IF NOT EXISTS\*/ `mysql` /\*!40100 DEFAULT CHARACTER SET latin1 \*/;
3. CREATE DATABASE /\*!32312 IF NOT EXISTS\*/ `userdb` /\*!40100 DEFAULT CHARACTER SET latin1 \*/;

步骤二：使用mysql 命令恢复删除的数据

以恢复userdb库为例，可参考下列操作。通常不建议直接覆盖旧库，而是采用建立新库并导入逻辑备份的方式执行恢复，待新库正常后即可废弃或删除旧库。

1）创建名为userdb2的新库

1. mysql> CREATE DATABASE userdb2;
2. Query OK, 1 row affected (0.00 sec)

2）导入备份文件，在新库中重建表及数据

1. [root@dbsvr1 ~]# mysql -u root -p userdb2 < /root/userdb.sql
2. Enter password:                                 //验证口令

3）确认新库正常，启用新库

1. mysql> USE userdb2;                             //切换到新库
2. Reading table information for completion of table and column names
3. You can turn off this feature to get a quicker startup with -A
4. Database changed
5. mysql> SELECT sn,username,uid,gid,homedir         //查询数据，确认可用
6. -> FROM userlist LIMIT 10;
7. +----+----------+-----+-----+-----------------+
8. | sn | username | uid | gid | homedir |
9. +----+----------+-----+-----+-----------------+
10. | 1 | root | 0 | 0 | /root |
11. | 2 | bin | 1 | 1 | /bin |
12. | 3 | daemon | 2 | 2 | /sbin |
13. | 4 | adm | 3 | 4 | /var/adm |
14. | 5 | lp | 4 | 7 | /var/spool/lpd |
15. | 6 | sync | 5 | 0 | /sbin |
16. | 7 | shutdown | 6 | 0 | /sbin |
17. | 8 | halt | 7 | 0 | /sbin |
18. | 9 | mail | 8 | 12 | /var/spool/mail |
19. | 10 | operator | 11 | 0 | /root |
20. +----+----------+-----+-----+-----------------+
21. 10 rows in set (0.00 sec)

4）废弃或删除旧库

1. mysql> DROP DATABASE userdb;
2. Query OK, 2 rows affected (0.09 sec)

**4 案例4：binlog日志**

**4.1 问题**

启用binlog日志，具体要求如下：

* 启用binlog日志，把日志文件存放到系统的/mylog目录下，日志文件为db50
* 手动创建3个新的日志文件
* 删除编号3之前的日志文件

**4.2 步骤**

实现此案例需要按照如下步骤进行。

步骤一：启用binlog日志

1）修改配置文件，并重启服务。

1. [root@dbsvr1 ~]# vim /etc/my.cnf
2. [mysqld]
3. server\_id=1 //指定server\_id
4. log-bin=/mylog/db50 //指定日志目录及名称
5. :wq
6. [root@dbsvr1 ~]# mkdir /mylog //创建目录
7. [root@dbsvr1 ~]# chown mysql /mylog //修改所有者
8. [root@dbsvr1 ~]# systemctl restart mysqld.service //重启服务

2）查看日志信息

1. [root@dbsvr1 ~]#
2. [root@localhost ~]# mysql -uroot -p123qqq...A //管理员登录
3. mysql: [Warning] Using a password on the command line interface can be insecure.
4. Welcome to the MySQL monitor. Commands end with ; or \g.
5. Your MySQL connection id is 3
6. Server version: 5.7.17-log MySQL Community Server (GPL)
7. Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.
8. Oracle is a registered trademark of Oracle Corporation and/or its
9. affiliates. Other names may be trademarks of their respective
10. owners.
11. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
12. mysql> show master status; //查看日志信息
13. +-------------+----------+--------------+------------------+-------------------+
14. | File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |
15. +-------------+----------+--------------+------------------+-------------------+
16. | db50.000001 | 154 | | | |
17. +-------------+----------+--------------+------------------+-------------------+
18. 1 row in set (0.00 sec)
19. mysql>

3）手动创建3个新的日志文件

1. mysql>
2. mysql> flush logs; //刷新日志
3. Query OK, 0 rows affected (0.14 sec)
4. mysql> flush logs; //刷新日志
5. Query OK, 0 rows affected (0.11 sec)
6. mysql> flush logs; //刷新日志
7. Query OK, 0 rows affected (0.12 sec)
8. mysql> system ls /mylog/ //查看日志文件
9. db50.000001 db50.000002 db50.000003 db50.000004 db50.index
10. mysql>
11. mysql> show master status; //查看日志信息
12. +-------------+----------+--------------+------------------+-------------------+
13. | File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |
14. +-------------+----------+--------------+------------------+-------------------+
15. | db50.000004 | 154 | | | |
16. +-------------+----------+--------------+------------------+-------------------+
17. 1 row in set (0.00 sec)
18. mysql>

4）删除编号3之前的日志文件

1. mysql>
2. mysql> purge master logs to "db50.000003"; //删除日志
3. Query OK, 0 rows affected (0.05 sec)
4. mysql> system ls /mylog/ //查看日志文件
5. db50.000003 db50.000004 db50.index
6. mysql>
7. mysql> system cat /mylog/db50.index //查看索引文件
8. /mylog/db50.000003
9. /mylog/db50.000004
10. mysql>

**5 案例5：使用binlog日志恢复数据**

**5.1 问题**

利用binlog恢复库表，要求如下：

* 启用binlog日志
* 创建db1库tb1表，插入3条记录
* 删除tb1表中刚插入的3条记录
* 使用mysqlbinlog恢复删除的3条记录

**5.2 步骤**

实现此案例需要按照如下步骤进行。

步骤一：启用binlog日志

1）调整/etc/my.cnf配置，并重启服务

1. [root@dbsvr1 ~]# vim /etc/my.cnf
2. [mysqld]
3. server\_id=1 //定义server\_id
4. log-bin=mysql-bin //定义日志名
5. binlog\_format=”mixed” //定义日志格式
6. [root@dbsvr1 ~]# systemctl restart mysqld.service //重启服务

2）确认binlog日志文件

新启用binlog后，每次启动MySQl服务都会新生成一份日志文件：

1. [root@dbsvr1 ~]# ls /var/lib/mysql/mysql-bin.\*
2. /var/lib/mysql/mysql-bin.000001 /var/lib/mysql/mysql-bin.index

其中mysql-bin.index文件记录了当前保持的二进制文件列表：

1. [root@dbsvr1 ~]# cat /var/lib/mysql/mysql-bin.index
2. ./mysql-bin.000001

重启MySQL服务程序，或者执行SQL操作“FLUSH LOGS;”，会生成一份新的日志：

1. [root@dbsvr1 ~]# ls /var/lib/mysql/mysql-bin.\*
2. /var/lib/mysql/mysql-bin.000001 /var/lib/mysql/mysql-bin.index
3. /var/lib/mysql/mysql-bin.000002
4. [root@dbsvr1 ~]# cat /var/lib/mysql/mysql-bin.index
5. ./mysql-bin.000001
6. ./mysql-bin.000002

步骤二：利用binlog日志重做数据库操作

1）执行数据库表添加操作

创建db1·库tb1表，表结构自定义：

1. mysql> CREATE DATABASE db1;
2. Query OK, 1 row affected (0.05 sec)
3. mysql> USE db1;
4. Database changed
5. mysql> CREATE TABLE tb1(
6. -> id int(4) NOT NULL,name varchar(24)
7. -> );
8. Query OK, 0 rows affected (0.28 sec)

插入3条表记录：

1. mysql> INSERT INTO tb1 VALUES
2. -> (1,'Jack'),
3. -> (2,'Kenthy'),
4. -> (3,'Bob');
5. Query OK, 3 rows affected (0.12 sec)
6. Records: 3 Duplicates: 0 Warnings: 0

确认插入的表记录数据：

1. mysql> SELECT \* FROM tb1;
2. +----+--------+
3. | id | name |
4. +----+--------+
5. | 1 | Jack |
6. | 2 | Kenthy |
7. | 3 | Bob |
8. +----+--------+
9. 3 rows in set (0.00 sec)

2）删除前一步添加的3条表记录

执行删除所有表记录操作：

1. mysql> DELETE FROM tb1;
2. Query OK, 3 rows affected (0.09 sec)

确认删除结果：

1. mysql> SELECT \* FROM tb1;
2. Empty set (0.00 sec)

步骤三：通过binlog日志恢复表记录

binlog会记录所有的数据库、表更改操作，所以可在必要的时候重新执行以前做过的一部分数据操作，但对于启用binlog之前已经存在的库、表数据将不适用。

根据上述“恢复被删除的3条表记录”的需求，应通过mysqlbinlog工具查看相关日志文件，找到删除这些表记录的时间点，只要恢复此前的SQL操作（主要是插入那3条记录的操作）即可。

1）查看mysql-bin.000002日志内容

1. [root@dbsvr1 ~]# mysqlbinlog /var/lib/mysql/mysql-bin.000002
2. /\*!50530 SET @@SESSION.PSEUDO\_SLAVE\_MODE=1\*/;
3. /\*!50003 SET @OLD\_COMPLETION\_TYPE=@@COMPLETION\_TYPE,COMPLETION\_TYPE=0\*/;
4. DELIMITER /\*!\*/;
5. # at 4
6. #170412 12:05:32 server id 1 end\_log\_pos 123 CRC32 0x6d8c069c Start: binlog v 4, server v 5.7.17-log created 170412 12:05:32 at startup
7. # Warning: this binlog is either in use or was not closed properly.
8. ROLLBACK/\*!\*/;
9. BINLOG '
10. jKftWA8BAAAAdwAAAHsAAAABAAQANS43LjE3LWxvZwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
11. AAAAAAAAAAAAAAAAAACMp+1YEzgNAAgAEgAEBAQEEgAAXwAEGggAAAAICAgCAAAACgoKKioAEjQA
12. AZwGjG0=
13. '/\*!\*/;
14. # at 123
15. #170412 12:05:32 server id 1 end\_log\_pos 154 CRC32 0x17f50164 Previous-GTIDs
16. # [empty]
17. # at 154
18. #170412 12:05:59 server id 1 end\_log\_pos 219 CRC32 0x4ba5a976 Anonymous\_GTID last\_committed=0 sequence\_number=1
19. SET @@SESSION.GTID\_NEXT= 'ANONYMOUS'/\*!\*/;
20. # at 219
21. #170412 12:05:59 server id 1 end\_log\_pos 310 CRC32 0x5b66ae13 Query thread\_id=3 exec\_time=0 error\_code=0
22. SET TIMESTAMP=1491969959/\*!\*/;
23. SET @@session.pseudo\_thread\_id=3/\*!\*/;
24. SET @@session.foreign\_key\_checks=1, @@session.sql\_auto\_is\_null=0, @@session.unique\_checks=1, @@session.autocommit=1/\*!\*/;
25. SET @@session.sql\_mode=1436549152/\*!\*/;
26. SET @@session.auto\_increment\_increment=1, @@session.auto\_increment\_offset=1/\*!\*/;
27. /\*!\C utf8 \*//\*!\*/;
28. SET @@session.character\_set\_client=33,@@session.collation\_connection=33,@@session.collation\_server=8/\*!\*/;
29. SET @@session.lc\_time\_names=0/\*!\*/;
30. SET @@session.collation\_database=DEFAULT/\*!\*/;
31. CREATE DATABASE db1
32. /\*!\*/;
33. # at 310
34. #170412 12:06:23 server id 1 end\_log\_pos 375 CRC32 0x2967cc28 Anonymous\_GTID last\_committed=1 sequence\_number=2
35. SET @@SESSION.GTID\_NEXT= 'ANONYMOUS'/\*!\*/;
36. # at 375
37. #170412 12:06:23 server id 1 end\_log\_pos 502 CRC32 0x5de09aae Query thread\_id=3 exec\_time=0 error\_code=0
38. use `db1`/\*!\*/;
39. SET TIMESTAMP=1491969983/\*!\*/;
40. CREATE TABLE tb1(
41. id int(4) NOT NULL,name varchar(24)
42. )
43. /\*!\*/;
44. # at 502
45. #170412 12:06:55 server id 1 end\_log\_pos 567 CRC32 0x0b8cd418 Anonymous\_GTID last\_committed=2 sequence\_number=3
46. SET @@SESSION.GTID\_NEXT= 'ANONYMOUS'/\*!\*/;
47. # at 567
48. #170412 12:06:55 server id 1 end\_log\_pos 644 CRC32 0x7e8f2fa0 Query thread\_id=3 exec\_time=0 error\_code=0
49. SET TIMESTAMP=1491970015/\*!\*/;
50. BEGIN
51. /\*!\*/;
52. # at 644
53. #170412 12:06:55 server id 1 end\_log\_pos 772 CRC32 0x4e3f728e Query thread\_id=3 exec\_time=0 error\_code=0 //插入表记录的起始时间点
54. SET TIMESTAMP=1491970015/\*!\*/;
55. INSERT INTO tb1 VALUES(1,'Jack'),(2,'Kenthy'), (3,'Bob')
56. /\*!\*/;
57. # at 772
58. #170412 12:06:55 server id 1 end\_log\_pos 803 CRC32 0x6138b21f Xid = 10
59. //确认事务的时间点
60. COMMIT/\*!\*/;
61. # at 803
62. #170412 12:07:24 server id 1 end\_log\_pos 868 CRC32 0xbef3f472 Anonymous\_GTID last\_committed=3 sequence\_number=4
63. SET @@SESSION.GTID\_NEXT= 'ANONYMOUS'/\*!\*/;
64. # at 868
65. #170412 12:07:24 server id 1 end\_log\_pos 945 CRC32 0x5684e92c Query thread\_id=3 exec\_time=0 error\_code=0
66. SET TIMESTAMP=1491970044/\*!\*/;
67. BEGIN
68. /\*!\*/;
69. # at 945
70. #170412 12:07:24 server id 1 end\_log\_pos 1032 CRC32 0x4c1c75fc Query thread\_id=3 exec\_time=0 error\_code=0 //删除表记录的时间点
71. SET TIMESTAMP=1491970044/\*!\*/;
72. DELETE FROM tb1
73. /\*!\*/;
74. # at 1032
75. #170412 12:07:24 server id 1 end\_log\_pos 1063 CRC32 0xccf549b2 Xid = 12
76. COMMIT/\*!\*/;
77. SET @@SESSION.GTID\_NEXT= 'AUTOMATIC' /\* added by mysqlbinlog \*/ /\*!\*/;
78. DELIMITER ;
79. # End of log file
80. /\*!50003 SET COMPLETION\_TYPE=@OLD\_COMPLETION\_TYPE\*/;
81. /\*!50530 SET @@SESSION.PSEUDO\_SLAVE\_MODE=0\*/;

2） 执行指定Pos节点范围内的sql命令恢复数据

根据上述日志分析，只要恢复从2014.01.12 20:12:14到2014.01.12 20:13:50之间的操作即可。可通过mysqlbinlog指定时间范围输出，结合管道交给msyql命令执行导入重做：

1. [root@dbsvr1 ~]# mysqlbinlog \
2. --start-datetime="2017-04-12 12:06:55" \
3. --stop-datetime="2017-04-12 12:07:23" \
4. /var/lib/mysql/mysql-bin.000002 | mysql -u root -p
5. Enter password:                                  //验证口令

3）确认恢复结果

1. mysql> SELECT \* FROM db1.tb1;
2. +----+--------+
3. | id | name |
4. +----+--------+
5. | 1 | Jack |
6. | 2 | Kenthy |
7. | 3 | Bob |
8. +----+--------+
9. 3 rows in set (0.00 sec)