NSD RDBM1 DAY04

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：

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

授权操作，此处可设置与从localhost访问时不同的密码：

```
mysql> GRANT all ON *.* TO root@'192.168.4.%' IDENTIFIED BY 'tarena';
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：

Top

```
[root@host120 ~]# mysql -u root -p -h 192.168.4.10
Enter password:
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 20
Server version: 5.7.17 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

从网络登入后，测试新建一个库、查看所有库：

```
mysql> CREATE DATABASE rootdb;                //创建新库rootdb
Query OK, 1 row affected (0.06 sec)

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| home               |
| mysql              |
| performance_schema |
| rootdb             |                    //新建的rootdb库
| sys                |
| userdb             |
+--------------------+
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：

```
mysql> GRANT all ON *.* TO dba007@localhost
    -> IDENTIFIED BY '123qqq…A '
    -> WITH GRANT OPTION;
```

Top

```
04.  Query OK, 0 rows affected (0.00 sec)
```

查看dba007的权限：

```
01.  mysql> SHOW GRANTS FOR dba007@localhost;
02.  +-----------------------------------------------------------------+
03.  | Grants for dba007@localhost                                     |
04.  +-----------------------------------------------------------------+
05.  | GRANT ALL PRIVILEGES ON *.* TO 'dba007'@'localhost' WITH GRANT OPTION |
06.  +-----------------------------------------------------------------+
07.  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：

```
01.  mysql> REVOKE all ON *.* FROM root@localhost;
02.  Query OK, 0 rows affected (0.00 sec)
03.  mysql> SHOW GRANTS FOR root@localhost;
04.  +-----------------------------------------------------+
05.  | Grants for root@localhost                           |
06.  +-----------------------------------------------------+
07.  | GRANT USAGE ON *.* TO 'root'@'localhost' WITH GRANT OPTION  |
08.  | GRANT PROXY ON ''@'' TO 'root'@'localhost' WITH GRANT OPTION |
09.  +-----------------------------------------------------+
10.  2 rows in set (0.00 sec)
```

验证撤销后的权限效果：

```
01.  mysql> exit                          //退出当前MySQL连接
02.  Bye
03.  [root@dbsvr1 ~]# mysql -u root -p          //重新以root从本地登入
04.  Enter password:
05.  Welcome to the MySQL monitor.  Commands end with ; or \g.
06.  Your MySQL connection id is 6                                    Top
07.  Server version: 5.6.15 MySQL Community Server (GPL)
08.
```

```
Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE newdb2014;                    //尝试新建库失败
ERROR 1044 (42000): Access denied for user 'root'@'localhost' to database 'newdb2
mysql> DROP DATABASE rootdb;                         //尝试删除库失败
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）：

```
mysql> GRANT all ON *.* TO root@localhost IDENTIFIED BY '1234567';
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：

```
mysql> exit                                          //退出当前MySQL连接
Bye
[root@dbsvr1 ~]# mysql -u dba007 -p                  //以另一个管理账号登入
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 24
Server version: 5.7.17 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

Top

由管理账号dba007重新为root添加本地访问权限：

```
mysql> GRANT all ON *.* TO root@localhost IDENTIFIED BY '1234567';
Query OK, 0 rows affected (0.00 sec)
mysql> SHOW GRANTS FOR root@localhost;            //查看恢复结果
+-----------------------------------------------------------------+
| Grants for root@localhost                                       |
+-----------------------------------------------------------------+
| GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH GRANT OPTION |
| GRANT PROXY ON ''@'' TO 'root'@'localhost' WITH GRANT OPTION    |
+-----------------------------------------------------------------+
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：

```
mysql> exit                                //退出当前MySQL连接
Bye
[root@dbsvr1 ~]# mysql -u root -p              //重新以root登入
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 25
Server version: 5.7.17 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE newdb2014;              //成功创建新库
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：

Top

```
mysql> GRANT all ON webdb.* TO webuser@'%' IDENTIFIED BY '888888';
```

```
02.  Query OK, 0 rows affected (0.00 sec)
```

查看授权结果：

```
01.  mysql> SHOW GRANTS FOR webuser@'%';
02.  +-----------------------------------------------+
03.  | Grants for webuser@%                          |
04.  +-----------------------------------------------+
05.  | GRANT USAGE ON *.* TO 'webuser'@'%'           |
06.  | GRANT ALL PRIVILEGES ON `webdb`.* TO 'webuser'@'%' |
07.  +-----------------------------------------------+
08.  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：

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

只赋予查询权限：

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

确认授权更改结果：

```
01.  mysql> SHOW GRANTS FOR webuser@'%';
02.  +---------------------------------------+
03.  | Grants for webuser@%                  |
04.  +---------------------------------------+
05.  | GRANT USAGE ON *.* TO 'webuser'@'%'   |
06.  | GRANT SELECT ON `webdb`.* TO 'webuser'@'%' |
07.  +---------------------------------------+
08.  2 rows in set (0.00 sec)
```

Top

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

```
[root@dbsvr1 ~]# systemctl  stop mysqld.service          //停止服务
[root@dbsvr1 ~]# systemctl  status mysqld.service         //确认状态
mysqld.service - MySQL Server
   Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled)
   Active: inactive (dead) since 五 2017-04-07 23:01:38 CST; 21s ago
     Docs: man:mysqld(8)
           http://dev.mysql.com/doc/refman/en/using-systemd.html
  Process: 20260 ExecStart=/usr/sbin/mysqld --daemonize --pid-file=/var/run/mysqld
  Process: 20238 ExecStartPre=/usr/bin/mysqld_pre_systemd (code=exited, status=0
 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：

```
[root@dbsvr1 ~]# vim /etc/my.cnf
[mysqld]
skip_grant_tables
.. ..
[root@dbsvr1 ~]# systemctl  start mysqld.service

[root@dbsvr1 ~]# service mysql status
mysqld.service - MySQL Server
   Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled)
   Active: active (running) since 五 2017-04-07 23:40:20 CST; 40s ago
     Docs: man:mysqld(8)
           http://dev.mysql.com/doc/refman/en/using-systemd.html
```

Top

```
13.      Process: 11698 ExecStart=/usr/sbin/mysqld --daemonize --pid-file=/var/run/mysqld
14.      Process: 11676 ExecStartPre=/usr/bin/mysqld_pre_systemd (code=exited, status=0
15.     Main PID: 11701 (mysqld)
16.       CGroup: /system.slice/mysqld.service
17.               └─11701 /usr/sbin/mysqld --daemonize --pid-file=/var/run/mysqld/mysqld.
```

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

```
01.  [root@dbsvr1 ~]# mysql  //直接回车即可
02.
03.  Welcome to the MySQL monitor.  Commands end with ; or \g.
04.  Your MySQL connection id is 4
05.  Server version: 5.7.17 MySQL Community Server (GPL)
06.
07.  Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.
08.
09.  Oracle is a registered trademark of Oracle Corporation and/or its
10.  affiliates. Other names may be trademarks of their respective
11.  owners.
12.
13.  Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
14.
15.  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：

```
01.  mysql> UPDATE mysql.user SET authentication_string=PASSWORD('123qqq…A')
02.      -> WHERE user='root' AND host='localhost';          //重设root的密码
03.  Query OK, 1 row affected, 1 warning (0.00 sec)
04.  Rows matched: 1  Changed: 1  Warnings: 1
05.  mysql> FLUSH PRIVILEGES;                          //刷新授权表
06.  Query OK, 0 rows affected (0.01 sec)
07.  mysql> exit                                    //退出mysql> 环境
08.  Bye
```

Top

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：

```
[root@dbsvr1 ~]# vim /etc/my.cnf
[mysqld]
#skip_grant_tables=1                    //注释掉或删除此行
.. ..
```

按正常方式，通过mysql脚本重启服务即可：

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

验证无密码登录时，将会被拒绝：

```
[root@dbsvr1 ~]# mysql -u root
Enter password:                    //没有跳过授权表回车会报错
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：

```
[root@dbsvr1 ~]# mysql -uroot –p123qqq…A
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.17 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Top

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：

```
[root@dbsvr1 ~]# mysqladmin -uroot -p password 'A…qqq321'
Enter password:                          //验证原来的密码
mysqladmin: [Warning] Using a password on the command line interface can be insecu
Warning: Since password will be sent to server in plain text, use ssl connection to ensu
[root@dbsvr1 ~]# mysql -uroot –pA…qqq321  //使用修改后的密码登录
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.17 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

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：

```
mysql> SET PASSWORD FOR root@localhost=PASSWORD('1234567');
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）：

Top

```
mysql> GRANT all ON *.* TO root@localhost IDENTIFIED BY '1234567';
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：

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

```
mysql> SET PASSWORD FOR root@localhost=PASSWORD('1234567');
ERROR 1290 (HY000): The MySQL server is running with the --skip-grant-tables option

mysql> GRANT all ON *.* TO root@localhost IDENTIFIED BY '1234567';
ERROR 1290 (HY000): The MySQL server is running with the --skip-grant-tables option
```

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：

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

Top

查看备份文件alldb.sql的部分内容：

```
[root@dbsvr1 ~]# grep -vE '^/|^-|^$' /root/alldb.sql | head -15
CREATE DATABASE /*!32312 IF NOT EXISTS*/ `home` /*!40100 DEFAULT CHARACTE
USE `home`;
DROP TABLE IF EXISTS `biao01`;
CREATE TABLE `biao01` (
  `id` int(2) NOT NULL,
  `name` varchar(8) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
LOCK TABLES `biao01` WRITE;
UNLOCK TABLES;
DROP TABLE IF EXISTS `biao02`;
CREATE TABLE `biao02` (
  `id` int(4) NOT NULL,
  `name` varchar(8) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
.. ..
```

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：

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

查看备份文件userdb.sql的部分内容：

```
[root@dbsvr1 ~]# grep -vE '^/|^-|^$' /root/userdb.sql
DROP TABLE IF EXISTS `stu_info`;
CREATE TABLE `stu_info` (
  `name` varchar(12) NOT NULL,
  `gender` enum('boy','girl') DEFAULT 'boy',
  `age` int(3) NOT NULL
```

Top

```
07.  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
08.  LOCK TABLES `stu_info` WRITE;
09.  .. ..
```

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：

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

查看备份文件userdb.sql的部分内容：

```
01.  [root@dbsvr1 ~]# grep '^CREATE DATA' /root/mysql+userdb.sql
02.  CREATE DATABASE /*!32312 IF NOT EXISTS*/ `mysql` /*!40100 DEFAULT CHARACTI
03.  CREATE DATABASE /*!32312 IF NOT EXISTS*/ `userdb` /*!40100 DEFAULT CHARAC
```

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

```
01.  mysql> CREATE DATABASE userdb2;
02.  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

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

3）确认新库正常，启用新库

```
01.  mysql> USE userdb2;                          //切换到新库
02.  Reading table information for completion of table and column names
03.  You can turn off this feature to get a quicker startup with -A
```

Top

```
04.
05.     Database changed
06.     mysql> SELECT sn,username,uid,gid,homedir          //查询数据，确认可用
07.         -> FROM userlist LIMIT 10;
08.     +----+----------+-----+-----+----------------+
09.     | sn | username | uid | gid | homedir        |
10.     +----+----------+-----+-----+----------------+
11.     |  1 | root     |   0 |   0 | /root          |
12.     |  2 | bin      |   1 |   1 | /bin           |
13.     |  3 | daemon   |   2 |   2 | /sbin          |
14.     |  4 | adm      |   3 |   4 | /var/adm       |
15.     |  5 | lp       |   4 |   7 | /var/spool/lpd |
16.     |  6 | sync     |   5 |   0 | /sbin          |
17.     |  7 | shutdown |   6 |   0 | /sbin          |
18.     |  8 | halt     |   7 |   0 | /sbin          |
19.     |  9 | mail     |   8 |  12 | /var/spool/mail |
20.     | 10 | operator |  11 |   0 | /root          |
21.     +----+----------+-----+-----+----------------+
22.     10 rows in set (0.00 sec)
```

4）废弃或删除旧库

```
01.     mysql> DROP DATABASE userdb;
02.     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。

Top

```
01.     [root@dbsvr1 ~]# vim  /etc/my.cnf
```

```
[mysqld]
    server_id=1  //指定server_id
log-bin=/mylog/db50  //指定日志目录及名称
:wq
[root@dbsvr1 ~]# mkdir  /mylog   //创建目录
[root@dbsvr1 ~]# chown  mysql  /mylog   //修改所有者
[root@dbsvr1 ~]# systemctl  restart mysqld.service  //重启服务
```

2）查看日志信息

```
[root@dbsvr1 ~]#
[root@localhost ~]# mysql -uroot -p123qqq...A //管理员登录
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.17-log MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show master status; //查看日志信息
+-------------+----------+--------------+------------------+-------------------+
| File        | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-------------+----------+--------------+------------------+-------------------+
| db50.000001 |      154 |              |                  |                   |
+-------------+----------+--------------+------------------+-------------------+
1 row in set (0.00 sec)

mysql>
```

3）手动创建3个新的日志文件

Top

```
mysql>
mysql> flush logs;  //刷新日志
```

```
Query OK, 0 rows affected (0.14 sec)

mysql> flush logs; //刷新日志
Query OK, 0 rows affected (0.11 sec)

mysql> flush logs; //刷新日志
Query OK, 0 rows affected (0.12 sec)

mysql> system ls /mylog/  //查看日志文件
db50.000001  db50.000002  db50.000003  db50.000004  db50.index
mysql>
mysql> show master status; //查看日志信息
+------------+----------+-------------+-----------------+-------------------+
| File       | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+------------+----------+-------------+-----------------+-------------------+
| db50.000004 |      154 |              |                  |                   |
+------------+----------+-------------+-----------------+-------------------+
1 row in set (0.00 sec)

mysql>
```

4）删除编号3之前的日志文件

```
 mysql>

mysql> purge  master  logs  to  "db50.000003"; //删除日志
Query OK, 0 rows affected (0.05 sec)

mysql> system ls /mylog/    //查看日志文件
db50.000003  db50.000004  db50.index
mysql>
mysql> system cat /mylog/db50.index //查看索引文件
/mylog/db50.000003
/mylog/db50.000004
mysql>
```

5 案例5：使用binlog日志恢复数据

Top

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

```
[root@dbsvr1 ~]# vim  /etc/my.cnf
[mysqld]
server_id=1  //定义server_id
log-bin=mysql-bin  //定义日志名
binlog_format=”mixed”     //定义日志格式
[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：

```
[root@dbsvr1 ~]# ls /var/lib/mysql/mysql-bin.*
/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：

```
[root@dbsvr1 ~]# cat /var/lib/mysql/mysql-bin.index
./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：

```
[root@dbsvr1 ~]# ls /var/lib/mysql/mysql-bin.*
/var/lib/mysql/mysql-bin.000001  /var/lib/mysql/mysql-bin.index
/var/lib/mysql/mysql-bin.000002

[root@dbsvr1 ~]# cat /var/lib/mysql/mysql-bin.index
./mysql-bin.000001
./mysql-bin.000002
```

Top

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：

```
mysql> CREATE DATABASE db1;
Query OK, 1 row affected (0.05 sec)

mysql> USE db1;
Database changed
mysql> CREATE TABLE tb1(
    -> id int(4) NOT NULL,name varchar(24)
    -> );
Query OK, 0 rows affected (0.28 sec)
```

插入3条表记录：

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

确认插入的表记录数据：

```
mysql> SELECT * FROM tb1;
+----+--------+
| id | name   |
+----+--------+
|  1 | Jack   |
|  2 | Kenthy |
|  3 | Bob    |
+----+--------+
3 rows in set (0.00 sec)
```

Top

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：

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

确认删除结果：

```
mysql> SELECT * FROM tb1;
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

```
[root@dbsvr1 ~]# mysqlbinlog /var/lib/mysql/mysql-bin.000002
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=1*/;
/*!50003 SET @OLD_COMPLETION_TYPE=@@COMPLETION_TYPE,COMPLETION_TYPE=
DELIMITER /*!*/;
# at 4
#170412 12:05:32 server id 1  end_log_pos 123 CRC32 0x6d8c069c  Start: binlog v
# Warning: this binlog is either in use or was not closed properly.
ROLLBACK/*!*/;
BINLOG '
jKftWA8BAAAAdwAAAHsAAAABAAQANS43LjE3LWxvZwAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAACMp+1YEzgNAAgAEgAEBAQEEgAAXwAEGggAAAAICAgCAAAACgoK
AZwGjG0=
'/*!*/;
# at 123
#170412 12:05:32 server id 1  end_log_pos 154 CRC32 0x17f50164  Previous-GTIDs
# [empty]
# at 154
#170412 12:05:59 server id 1  end_log_pos 219 CRC32 0x4ba5a976  Anonymous_G
SET @@SESSION.GTID_NEXT= 'ANONYMOUS'/*!*/;
# at 219
#170412 12:05:59 server id 1  end_log_pos 310 CRC32 0x5b66ae13  Query  thread
SET TIMESTAMP=1491969959/*!*/;
SET @@session.pseudo_thread_id=3/*!*/;
```

top

```
SET @@session.foreign_key_checks=1, @@session.sql_auto_is_null=0, @@session.ur
SET @@session.sql_mode=1436549152/*!*/;
SET @@session.auto_increment_increment=1, @@session.auto_increment_offset=1/*
/*!\C utf8 *//*!*/;
SET @@session.character_set_client=33,@@session.collation_connection=33,@@sess
SET @@session.lc_time_names=0/*!*/;
SET @@session.collation_database=DEFAULT/*!*/;
CREATE DATABASE db1
/*!*/;
# at 310
#170412 12:06:23 server id 1  end_log_pos 375 CRC32 0x2967cc28  Anonymous_G
SET @@SESSION.GTID_NEXT= 'ANONYMOUS'/*!*/;
# at 375
#170412 12:06:23 server id 1  end_log_pos 502 CRC32 0x5de09aae  Query   thread_
use `db1`/*!*/;
SET TIMESTAMP=1491969983/*!*/;
CREATE TABLE tb1(
id int(4) NOT NULL,name varchar(24)
)
/*!*/;
# at 502
#170412 12:06:55 server id 1  end_log_pos 567 CRC32 0x0b8cd418  Anonymous_G
SET @@SESSION.GTID_NEXT= 'ANONYMOUS'/*!*/;
# at 567
#170412 12:06:55 server id 1  end_log_pos 644 CRC32 0x7e8f2fa0  Query   thread_
SET TIMESTAMP=1491970015/*!*/;
BEGIN
/*!*/;
# at 644
#170412 12:06:55 server id 1  end_log_pos 772 CRC32 0x4e3f728e  Query   thread_
SET TIMESTAMP=1491970015/*!*/;
INSERT INTO tb1 VALUES(1,'Jack'),(2,'Kenthy'), (3,'Bob')
/*!*/;
# at 772
#170412 12:06:55 server id 1  end_log_pos 803 CRC32 0x6138b21f  Xid = 10
                                        //确认事务的时间点
COMMIT/*!*/;
# at 803
#170412 12:07:24 server id 1  end_log_pos 868 CRC32 0xbef3f472  Anonymous_GT
SET @@SESSION.GTID_NEXT= 'ANONYMOUS'/*!*/;
# at 868
```

Top

```
#170412 12:07:24 server id 1  end_log_pos 945 CRC32 0x5684e92c  Query   thread
SET TIMESTAMP=1491970044/*!*/;
BEGIN
/*!*/;
# at 945
#170412 12:07:24 server id 1  end_log_pos 1032 CRC32 0x4c1c75fc        Query   th
SET TIMESTAMP=1491970044/*!*/;
DELETE FROM tb1
/*!*/;
# at 1032
#170412 12:07:24 server id 1  end_log_pos 1063 CRC32 0xccf549b2        Xid = 12
COMMIT/*!*/;
SET @@SESSION.GTID_NEXT= 'AUTOMATIC' /* added by mysqlbinlog */ /*!*/;
DELIMITER ;
# End of log file
/*!50003 SET COMPLETION_TYPE=@OLD_COMPLETION_TYPE*/;
/*!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：

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

3）确认恢复结果

```
mysql> SELECT * FROM db1.tb1;
+----+--------+
| id | name   |
+----+--------+
|  1 | Jack   |
|  2 | Kenthy |
|  3 | Bob    |
+----+--------+
```

Top

```
09.     3 rows in set (0.00 sec)
```

Top