目录

[一、什么是主从复制 1](#_Toc432414398)

[二、主从复制的原理 1](#_Toc432414399)

[三、如何授权一个用户： 2](#_Toc432414400)

[四、如何开启log-bin日志 3](#_Toc432414401)

[1、开启log-bin日志 3](#_Toc432414402)

[2、和log-bin日志相关的几个函数 4](#_Toc432414403)

[3、查看log-bin日志文件里面的内容： 4](#_Toc432414404)

[4、案例：使用log-bin日志完成数据的回复 6](#_Toc432414405)

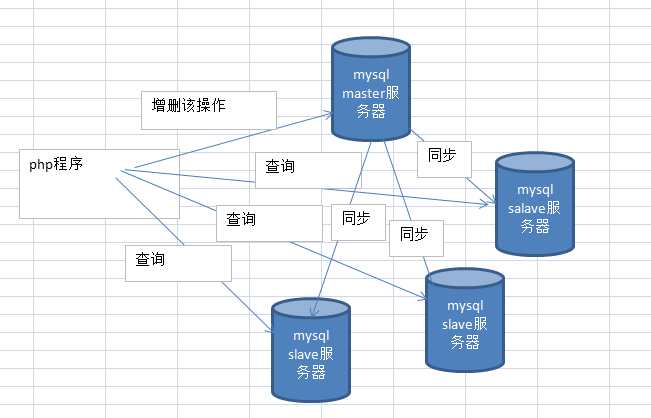
[五、主从配置 8](#_Toc432414406)

[1、主服务器的配置（xp1）192.168.28.251 8](#_Toc432414407)

[2、从服务器的配置(xp2) 192.168.28.252 9](#_Toc432414408)

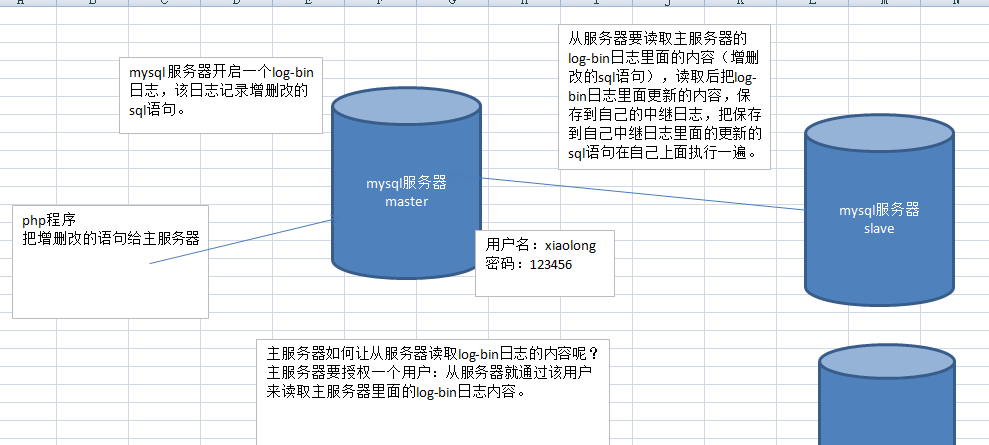
# 一、什么是主从复制

至少需要2台mysql服务器，一台配置为主服务器，一台配置为从服务器，主服务器的数据要同步到从服务器。



# 二、主从复制的原理

mysql中有一种日志，叫做bin日志（二进制日志），会记录下所有修改过数据库的sql语句。主从复制的原理实际是多台服务器都开启bin日志，然后主服务器会把执行过的sql语句记录到bin日志中，之后把这个bin日志发给从服务器，在从服务器再把bin日志中记录的sql语句同样的执行一遍。这样从服务器上的数据就和主服务器相同了。



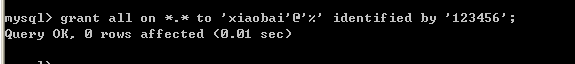
实现方式：（1）授权一个用户，（2）开启log-bin日志，（3）详细的配置过程。

# 三、如何授权一个用户：

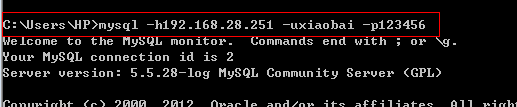
（1）mysql添加账号：

语法：grant 权限 on 指定数据库和表 to “用户名”@’可以登录主机的ip地址’ identified by ‘密码’

比如：grant all on \*.\* to ‘xiaolong’@’%’ identified by ‘123456’;



对授权的用户远程登录：



（2）删除账号：

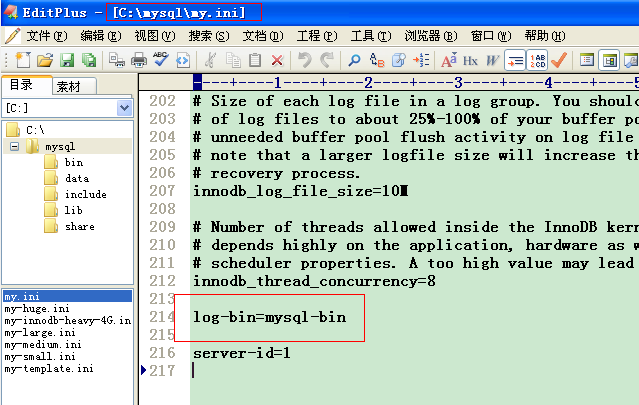
语法：drop user ‘用户名’@’授权登录主机的ip地址’



# 四、如何开启log-bin日志

## 1、开启log-bin日志

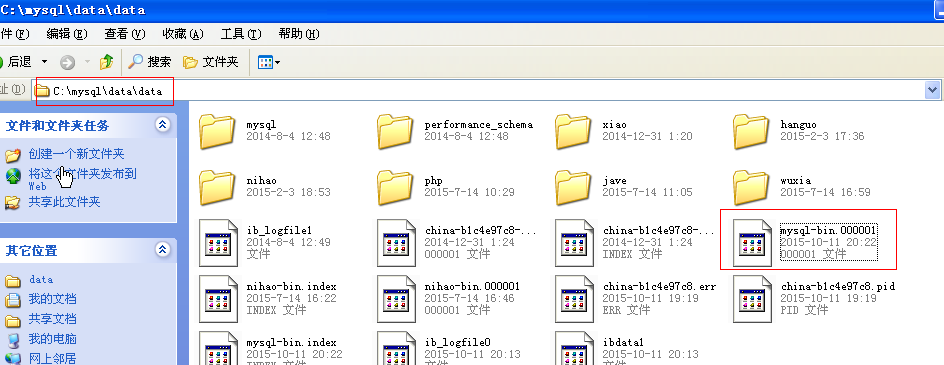
打开mysql的配置文件：my.ini(window) my.cnf (linux)



mysql-bin是二进制日志文件的名称，默认是和数据库文件同一级目录里面存储。在此处也可以自己指定存储位置，比如log-bin=d:/nihao.txt

注意：log-bin日志，会在mysql服务重启时，会产生新的一个日志文件。

新产生的log-bin日志文件如下：



## 2、和log-bin日志相关的几个函数

（1）reset master

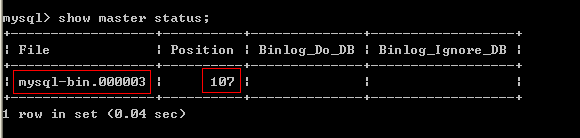
清空所有的 log-bin日志，并产生一个新的log-bin日志文件。

（2）flush logs

产生新的一个log-bin日志文件。

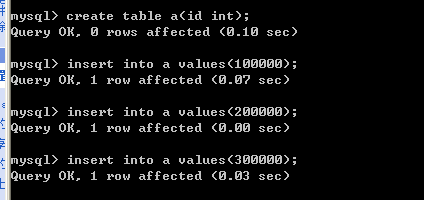
（3）show master status

查看最新的一个log-bin日志的文件名称，并包含pos位置。



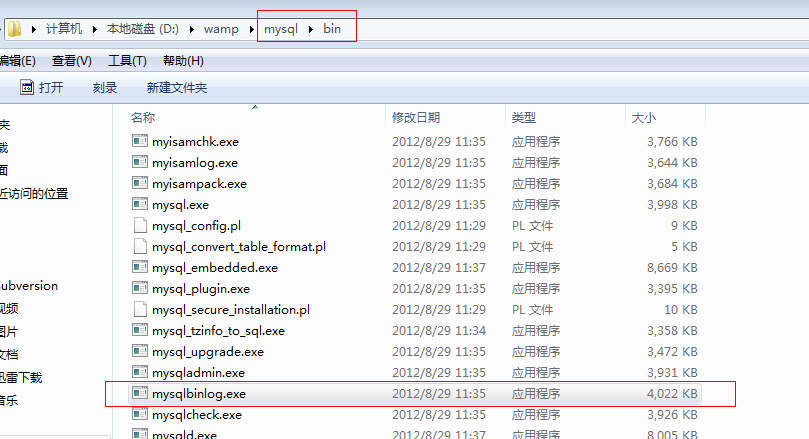
## 3、查看log-bin日志文件里面的内容：

创建一个表，添加sql语句进行测试：



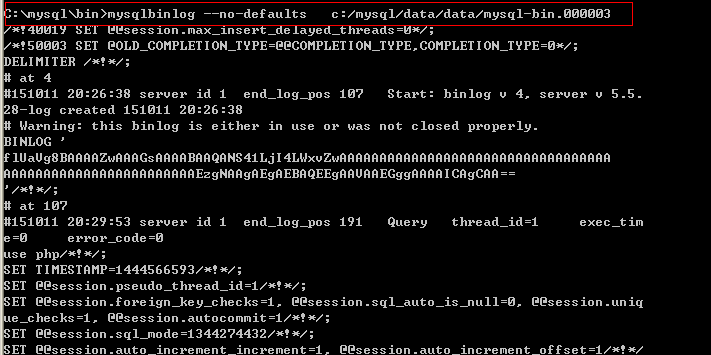
如何查看log-bin日志的内容呢？

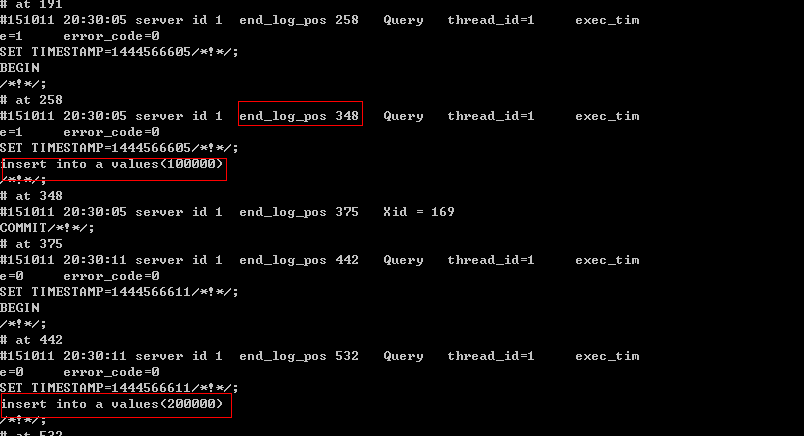
使用一个命令：在mysql的安装目录的bin目录下面的mysqlbinlog.exe命令来完成查看。



语法：通过cmd的方式，进入到bin 目录里面执行。

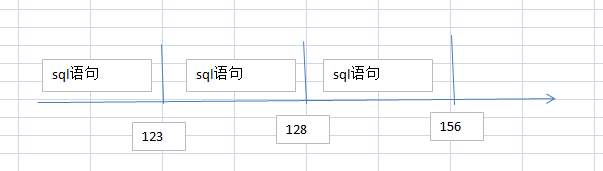
mysqlbinlog --no-defaults 日志文件的名称（包含全路径）





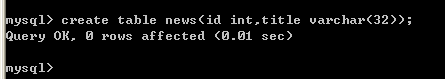
pos位置，记录sql语句在log-bin日志文件中的位置。

pos所在位置的分析：end\_log\_pos :记录上一个 sql语句的结束，下一个 sql语句的开始

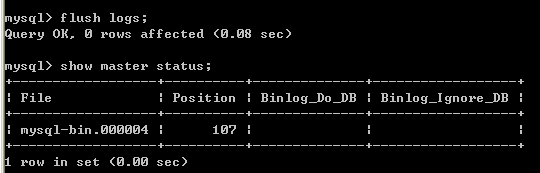


## 4、案例：使用log-bin日志完成数据的回复

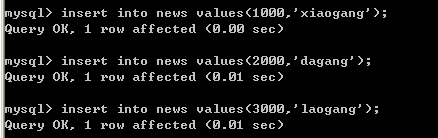
（1）创建一张表



（2）执行flush logs（产生新的一个日志文件）

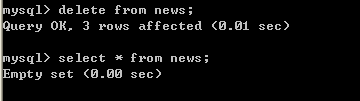


（3）使用insert语句完成数据的插入



（4）执行flush logs（产生新的一个日志文件）

（5）把该表里面的数据给删除。



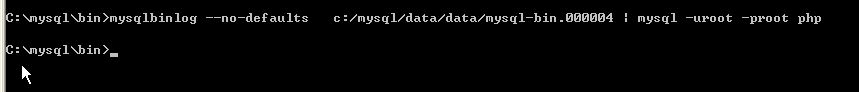
（6）完成log-bin日志恢复数据。

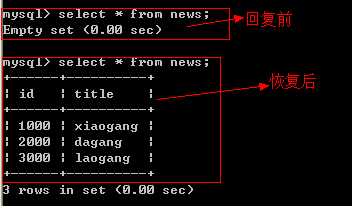
要注意：刚才insert的操作是被记录到mysql-bin.000004日志文件里面的。

只需要，把mysql-bin.000004日志文件里面的sql语句执行一遍，即可完成 数据恢复。

语法：

mysqlbinlog --no-defaults 日志文件的名称（路径）| mysql –uroot –proot 数据库的名称





案例扩展：

比如一个公司在上午9:00备份了一次数据，到9:30分的时候，由于员工的误操作，把所有的数据给弄丢了，要求恢复数据。

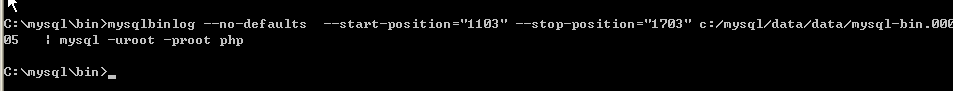
分析：上午9:00之前的数据，非常好恢复的，因为已经备份过。关键是以从9：00到9:30的数据，如何恢复。

思路：查看log-bin日志文件，把9：00到9:30之间的增删改的sql语句执行一遍即可。

查找记录增删改sql语句在log-bin日志文件中的pos 开始位置和结束位置。

语法：

使用：--start-position=”开始位置” --stop-position=”结束的位置”

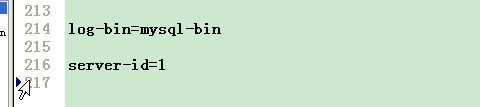


# 五、主从配置

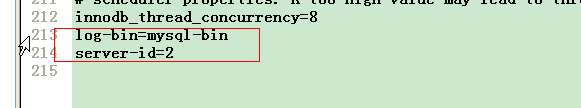
## 1、主服务器的配置（xp1）192.168.28.251

（1）主从服务器都要开启log-bin日志，并设置一个不同的server-id 的值。

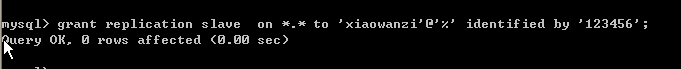
主服务器的配置：

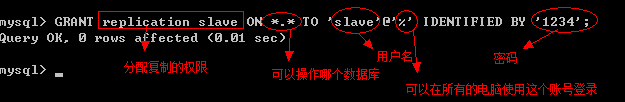


从服务器的配置：



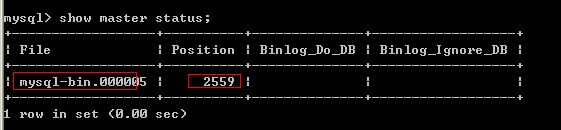
（2）在主服务器上面授权一个用户，从服务器就是通过该账号，完成读取 log-bin日志信息的。





（3）查看主服务器上面的最新的log-bin 日志，

注意：此时，就禁止对主服务器进行更改的操作。



## 2、从服务器的配置(xp2) 192.168.28.252

（1）先关闭从服务器（每次重新配置时需要先关闭）

执行命令：stop slave

（2）开始配置：

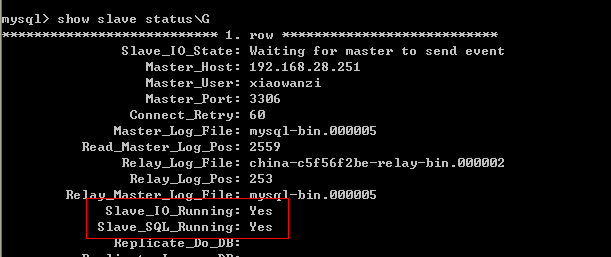
change master to master\_host=‘主服务器的ip地址’,master\_user=‘主服务器上用于同步数据的账号’,master\_password=‘同步账号的密码’,master\_log\_file=‘bin日志的文件名’,master\_log\_pos=bin日志中的position值。



（3）开启从服务器

执行命令：start slave

（4）查看从服务器的状态，是否配置成功



Slave\_IO\_Running:Yes

此进程负责从服务器从主服务器上读取binlog 日志，并写入从服务器上的中继日志。

Slave\_SQL\_Running:Yes

此进程负责读取并且执行中继日志中的binlog日志，

注：以上两个都为yes则表明成功，只要其中一个进程的状态是no，则表示复制进程停止，错误原因可以从”last\_error”字段的值中看到。

通过在主服务器上面创建一个新的数据库，并创建一张新表，并添加记录，查看从服务器是否同步来完成测试。

配置完成后，要禁止对从服务器执行增删改的操作。