# 实验五 数据库的备份与恢复

## 实验目的

在课程理论知识学习的基础上，使用MySQL数据库管理系统工具对数据库的备份恢复操作有直接的体会。

## 实验内容

在生产环境中，数据库中的数据可能会遭遇以下各种不测从而导致数据丢失：

* 硬件故障
* 软件故障
* 自然灾害
* 黑客攻击
* 误操作

为了能够及时恢复数据，有必要做好数据的备份工作。一旦发生故障，可以及时恢复数据。数据的备份类型根据其自身的特性主要分为以下几种：

* 完全备份：备份整个数据集(即整个数据库)。
  + 优点：备份保持最新备份，恢复时可以一次恢复，花费时间少；
  + 缺点：如果数据量大，长时间的备份操作会对系统造成负载过重。
* 部分备份：备份部分数据集(例如：只备份一个表)。部分备份又可细分为增量备份和差异备份。
  + 增量备份：备份自上一次备份以来(增量或完全)以来变化的数据，该备份节约空间、还原麻烦、需要保证前一次的备份数据可恢复。
  + 差异备份：备份自上一次完全备份以来变化的数据，该备份浪费空间、还原比增量备份简单。

根据对数据的备份操作方法不同，备份主要有物理备份和逻辑备份：

* 物理备份：转存数据库的物理文件，比如数据文件、控制文件、归档日志文件等。通过拷贝等命令直接打包复制数据库的数据文件达到备份的效果。
  + 优点：备份简单、恢复简单、备份的速度快
  + 缺点：不是那么的容易跨平台、备份的文件比较大
* 逻辑备份：通过逻辑导出对数据进行备份。一般是通过特定工具从数据库中导出数据并另存备份。常见方式mysqldump。

备份数据的策略要根据不同的应用场景进行定制，大致有几个参考因素，我们可以根据这些因素定制符合特定环境中的数据备份策略：

* 能够容忍丢失多少数据
* 恢复数据需要多长时间
* 需要恢复哪一些数据

在数据库存储引擎一致的情况下，一般来说，如果数据量较小，可以使用物理备份，直接复制数据库文件；如果数据量一般，可以使用mysqldump对数据库进行完全备份，然后定期备份BINARY LOG达到增量备份的效果；如果数据量一般，而又不过分影响业务运行，可以使用lvm2的快照对数据文件进行备份，而后定期备份BINARY LOG达到增量备份的效果；如果数据量很大，而又不过分影响业务运行，可以使用xtrabackup进行完全备份后，定期使用xtrabackup进行增量备份或差异备份。

以下将在了解MySQL数据库管理系统中的数据库备份、恢复机制的基础上，进行有关MySQL数据库管理系统的备份和恢复操作。主要实践三种不同的备份策略：

1. 逻辑备份：通过mysqldump进行全库备份与恢复
2. 逻辑备份：通过mysqldump+binlog进行备份与完整恢复
3. 逻辑备份：通过mysqldump+binlog进行增量备份与基于pos位置的恢复

## 实验环境

普通PC、Windows系列操作系统、MySQL 数据库管理系统

## 实验步骤

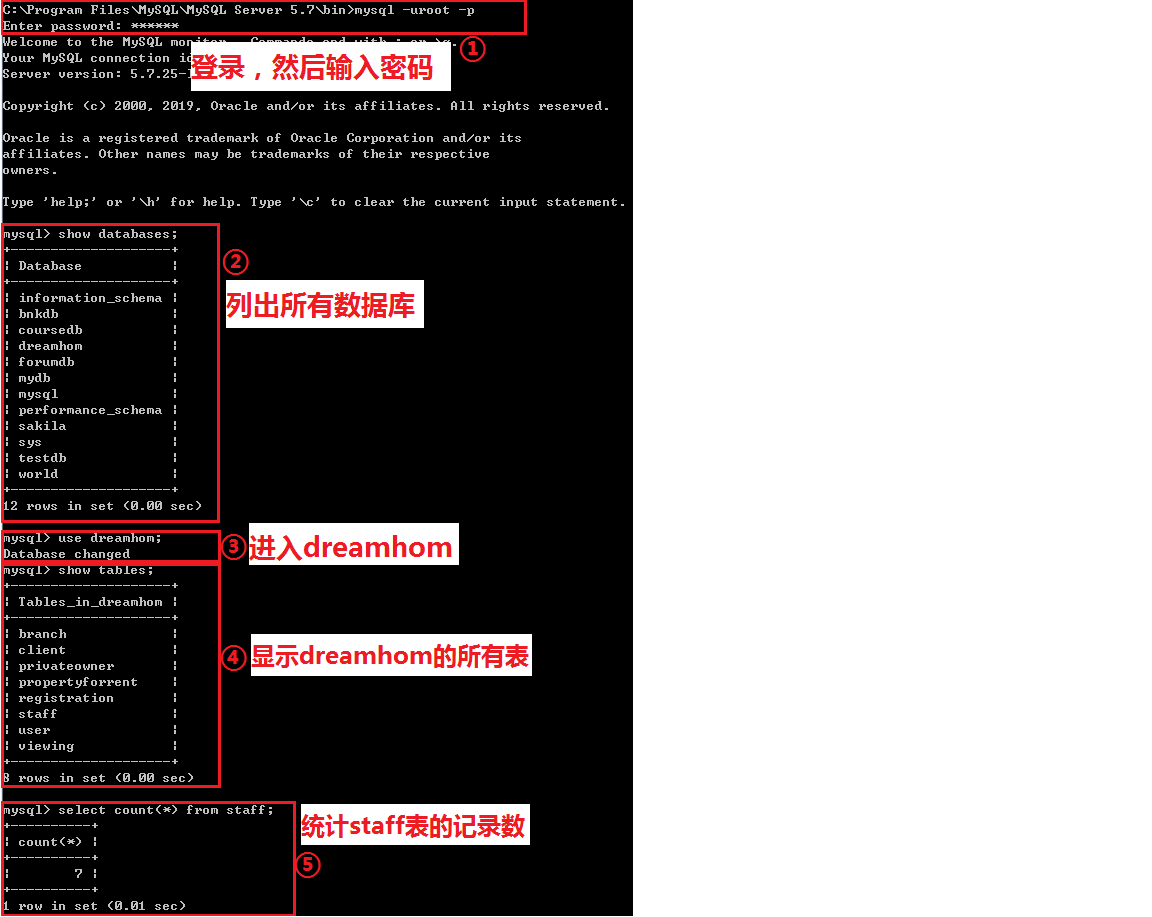
1. 逻辑备份：通过mysqldump进行全库备份与恢复。

操作的数据库：dreamhom

操作环境：MySQL Server的命令行模式

模拟场景：误删数据库

步骤一：查看数据库以及当前数据库中的表和数据。



步骤二：执行如下命令，对数据库进行完全备份：

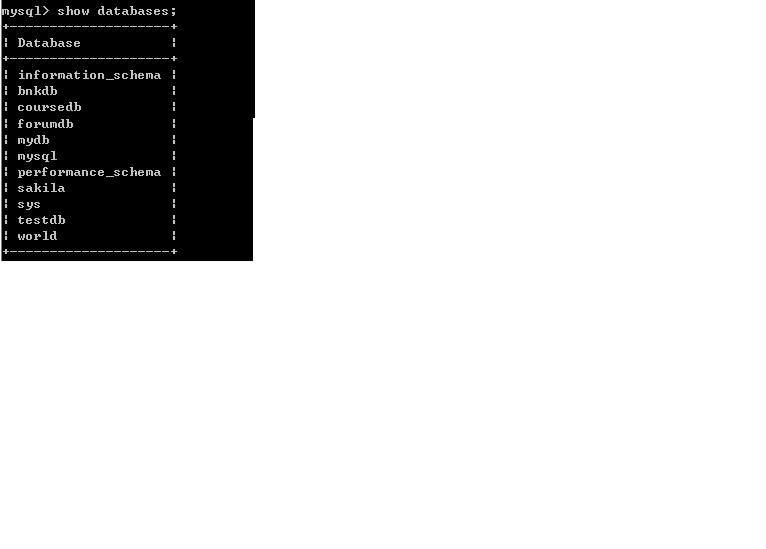


上面这行代码功能是将dreamhom数据库全量备份。其中备份的路径为当前路径，备份的文件名称为backup001.sql（可自行命名）。

* 参数-flush-logs：使用一个新的日志文件记录接下来的日志
* 参数-lock-all-tables：锁定所有数据库表

回到刚刚备份的文件夹中可以看到新生成的文件backup001.sql。这里的文件夹为C:\Program Files\MySQL\MySQL Server 5.7\bin\。

步骤三：模拟数据库丢失。回到MySQL Server中，把dreamhom数据库删除掉。使用命令：drop database dreamhom。重新查看数据库，已经看不到dreamhom。



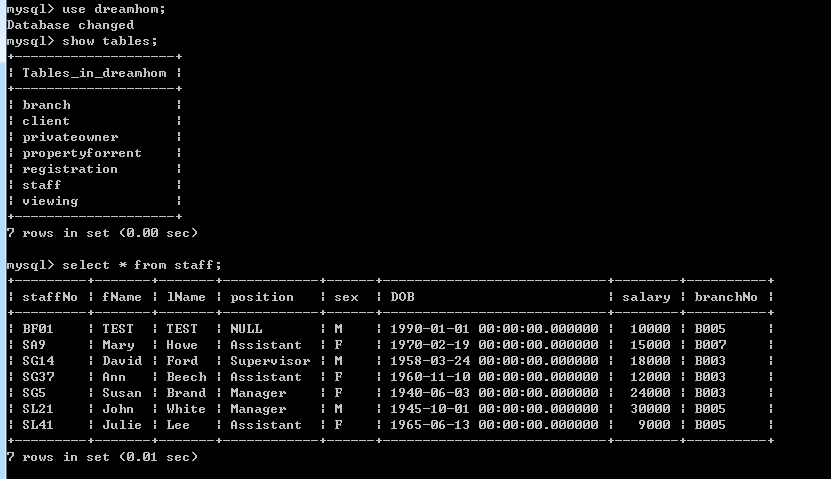
步骤四：重新创建一个数据库dreamhom。Create database dreamhom;查看该库中的表，返回结果为空。



步骤五：利用导出的备份文件backup001.sql恢复数据库以及数据：



步骤六：验证是否恢复成功。重新进入dreamhom数据库查看表和数据：



1. 逻辑备份：通过mysqldump+binlog进行增量备份与完整恢复

操作的数据库：dreamhom

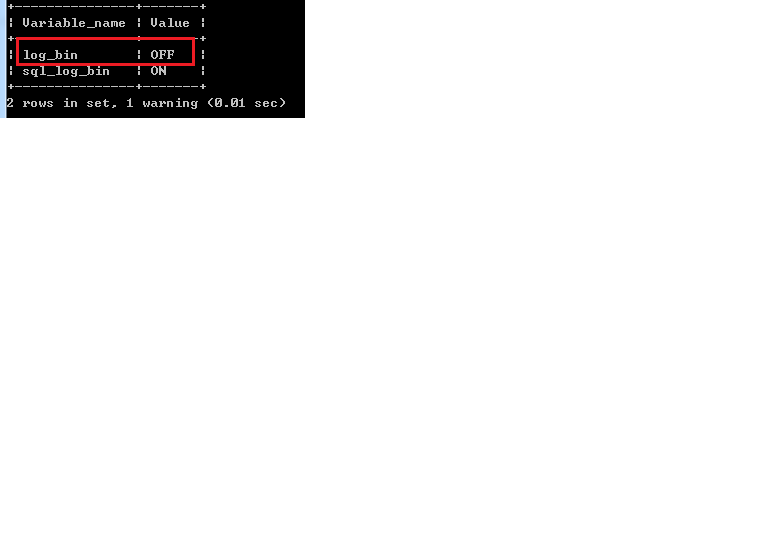
操作环境：MySQL Server的命令行模式

模拟场景：误删数据库中的数据并且数据备份在单独的日志文件中

步骤一：打开备份日志开关。在进行增量备份之前需要查看一下配置文件，查看 log\_bin 是否开启，因为要做增量备份首先要开启 log\_bin 。首先，登录到MySQL命令行模式，输入如下命令：



看到如下图所示的执行结果，表示未开启：



修改MySQL配置文件，找到log-bin，将其修改为：

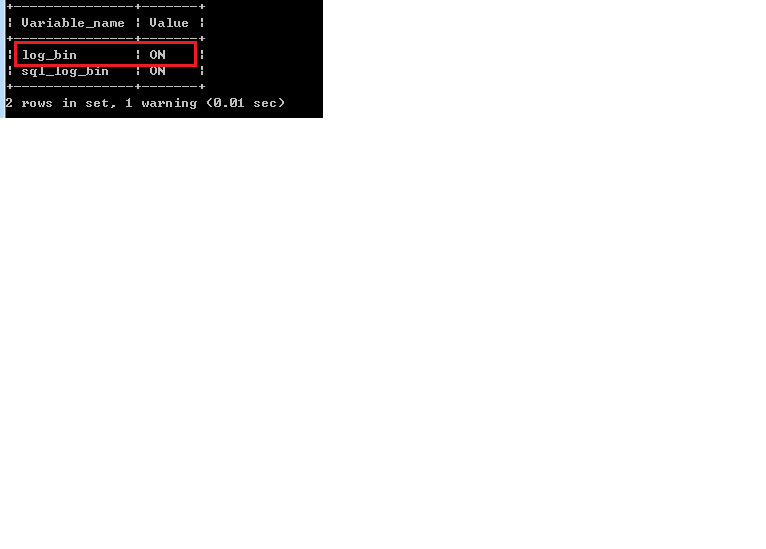
log-bin = log-bin = C:\ProgramData\MySQL\mysql-log-bin

* win7系统中，MySQL配置文件名为my.ini，默认在C:\ProgramData\MySQL\MySQL Server 5.7路径下。
* mysql-log-bin是为后面的日志文件创建的默认文件名开头部分。

修改之后重启MySQL服务，再次输入：



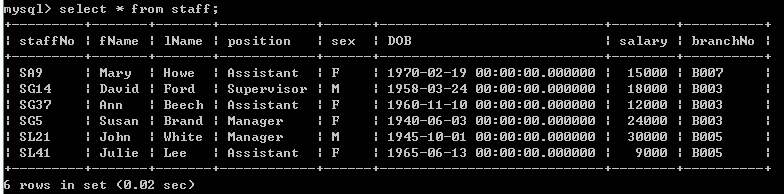
此时状态如下图，表示日志已经打开。



步骤二：查看并记录当前使用的日志文件以及数据库表的状态。使用如下命令查看日志文件。说明当前正在记录日志的文件名字为mysql-log-bin.000001。



当前数据库中有如下数据：

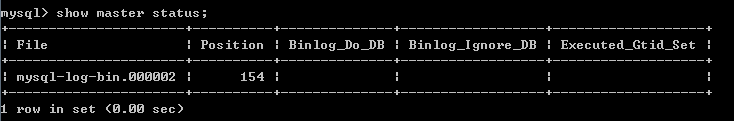


步骤三：插入一条数据：INSERT INTO `dreamhom`.`staff` (`staffNo`, `fName`, `lName`, `sex`, `DOB`, `salary`, `branchNo`) VALUES ('BF01', 'TEST', 'TEST', 'M', '1990-01-01 ', '10000', 'B005');

步骤四：执行如下命令，启用新的日志文件，此后，对数据库做的所有操作都将记录在新的日志文件内。

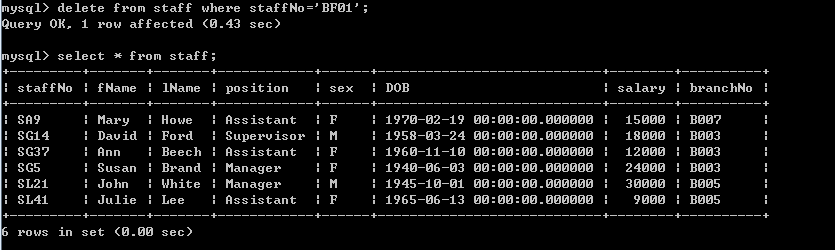


执行完之后，没有提示。重新进入MySQL，执行命令，查看当前的日志文件：

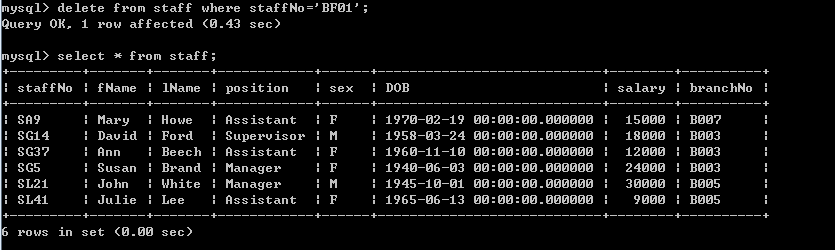


可看到日志文件从mysql-log-bin.000001变为mysql-log-bin.000002。到这里为止，增量备份已经完成。

步骤五：模拟数据丢失场景。将刚刚插入的数据删除：



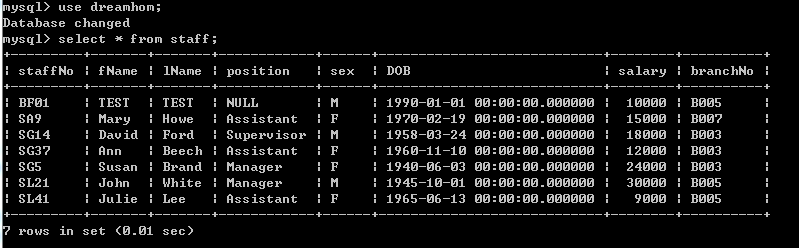
查询staff表，数据只有六条：



步骤六：恢复数据。尝试从mysql-log-bin.000001中恢复数据：



这里的C:\ProgramData\MySQL\是日志文件存储的文件夹，就是步骤一中设置的日志文件存储路径。执行完之后，没有提示，重新进入数据库，查看数据：



被删除的staffNo为BF01的记录已经恢复回来。到此数据恢复完成。

1. 逻辑备份：通过mysqldump+binlog进行增量备份与基于position的恢复

操作的数据库：dreamhom

操作环境：MySQL Server的命令行模式

模拟场景：误删数据库并且所有的操作在同一个日志文件中

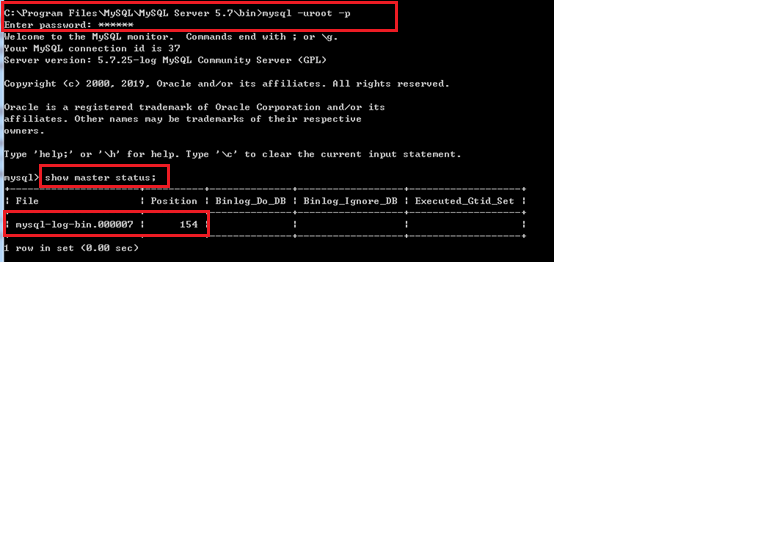
步骤一：进行完全备份。执行如下命令完整备份一次数据库，备份到当前目录的backup20190401003.sql文件中。



参数说明：

* -B：备份时添加建库语句
* -l：数据库表添加只读锁
* -F：刷新bin-log日志文件，相当于执行flush logs指令，生成新的bin-log日志文件，此后日志都放在新的bin-log日志文件中，利于查找。

步骤二：登录进入MySQL server，查看当前使用的bin-log日志文件。当前使用的日志文件是myslq-log-bin.000007，记录的位置是154。



创建新表并写入一条数据（这一条数据请用自己的学号和姓名全拼写入）：

CREATE TABLE `dreamhom`.`user` (

`id` INT NOT NULL,

`name` VARCHAR(20) NULL,

PRIMARY KEY (`id`));

INSERT INTO `dreamhom`.`user` (`id`, `name`) VALUES ('1', 'lijingjing');

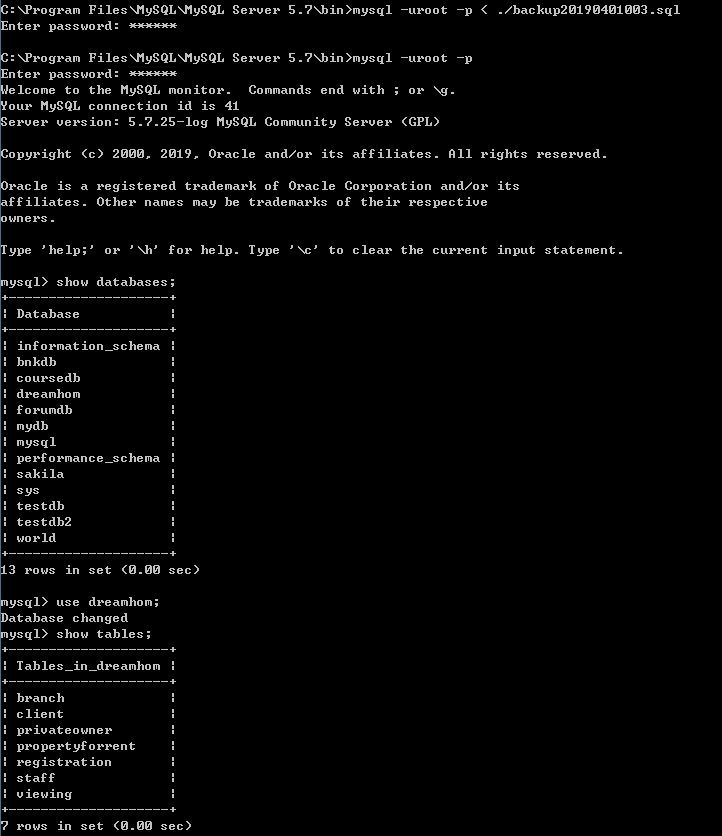
步骤三：再一次查看当前使用的binlog日志文件并记住position的值：



步骤四：模拟数据库丢失。删除数据库：



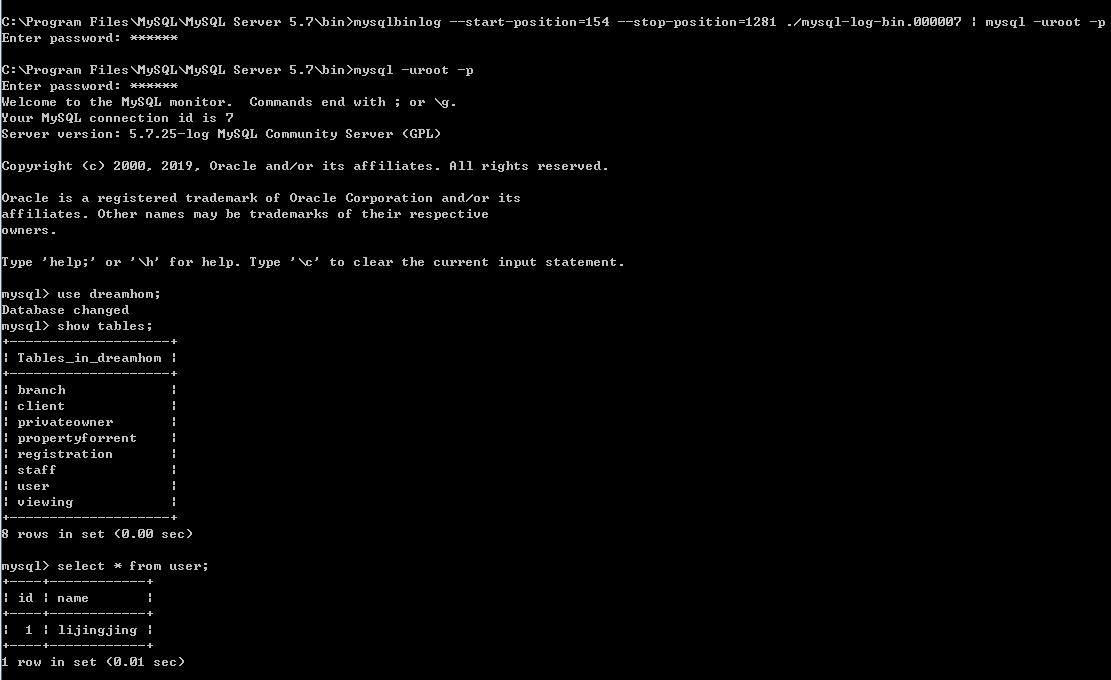
步骤五：数据库恢复。恢复数据库需要两个步骤，其一，从完全备份恢复，其二，从日志文件中获得增量备份。首先进行完全备份的恢复。



登录进入MySQL Server，查看dreamhom数据库以及表。

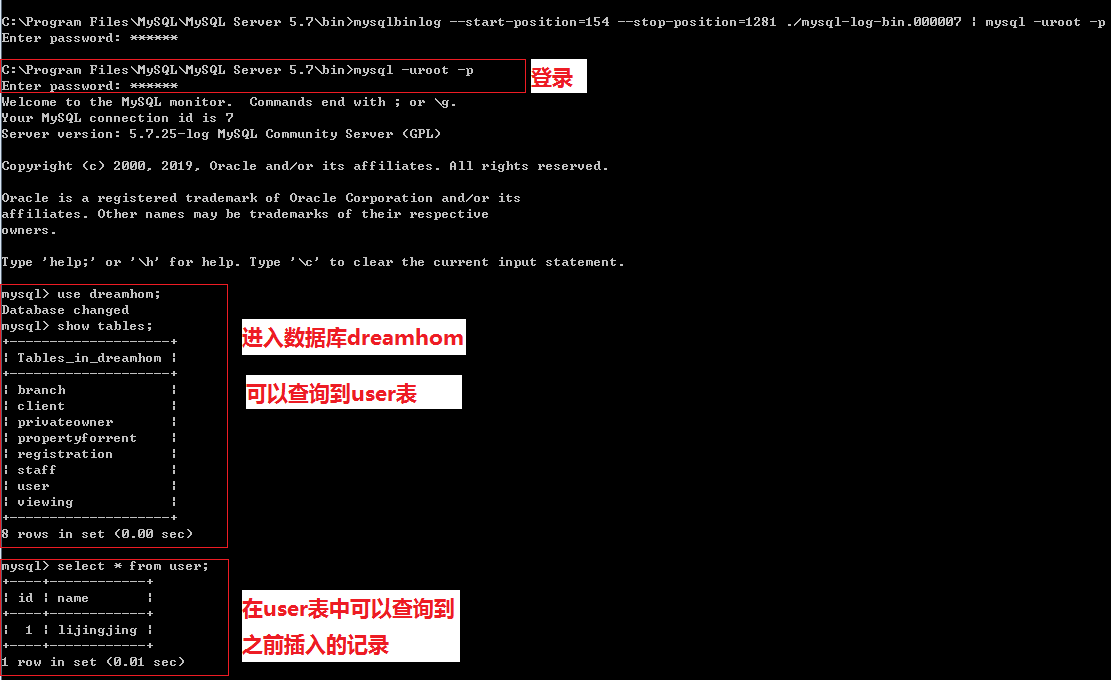


从上图中可以看到，恢复的完全备份中不包含新建的表以及插入的数据。这一部分需要从日志文件中进行增量恢复。



* --start-position：指定恢复开始的pos值位置，这里的154是步骤二中记录的值；
* --stop-position： 指定恢复结束的pos值位置，这里的1281是步骤三中记录的值；
* mysql-log-bin.000007是当下正在使用的日志文件名字。

步骤六：验证是否恢复成功。登录进入MySQL Server检验恢复是否成功。



到此为止，恢复成功。

## 注意事项

1. 以上操作使用MySQL Server自带的命令行模式进行操作。
2. 在以上操作的第三种备份恢复策略中，请大家先记住position的值，也就是我们想要恢复的日志操作的位置。但是，在生产环境中，通常没有办法预料故障出现的位置。因此，position的值是没有办法提前预知的。那么，一旦出现了故障，应该怎样来查找position的值？请写出查找position值的详细方法（有图以及文字描述）。
3. 除去以上的备份和恢复方式，在生产环境中，我们还时常面临基于时间进行数据恢复的状况，也就是恢复某一个时间段的数据，在上述实验的第三种备份和恢复策略中，稍作修改可以实现基于时间段的数据恢复，请问如何操作？给出详细的操作步骤（有图以及文字描述）。
4. 以上1）自行按照步骤操作完成，不需要写入实验报告。2）3）请写入实验报告。