**北京邮电大学软件学院**

**2019-2020学年第2学期实验报告**

**课程名称： 数据库系统原理课程设计\_\_\_\_\_\_**

**实验名称：**  实验五 数据库的备份与恢复

**实验完成人：**

**姓名： 王衔飞 学号：**

**指导教师：**\_\_\_\_\_\_\_\_\_ **刁婷** \_\_\_\_\_\_\_\_\_\_\_

**日 期： 2019 年 5 月 20 日**

1. **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。

1. **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：

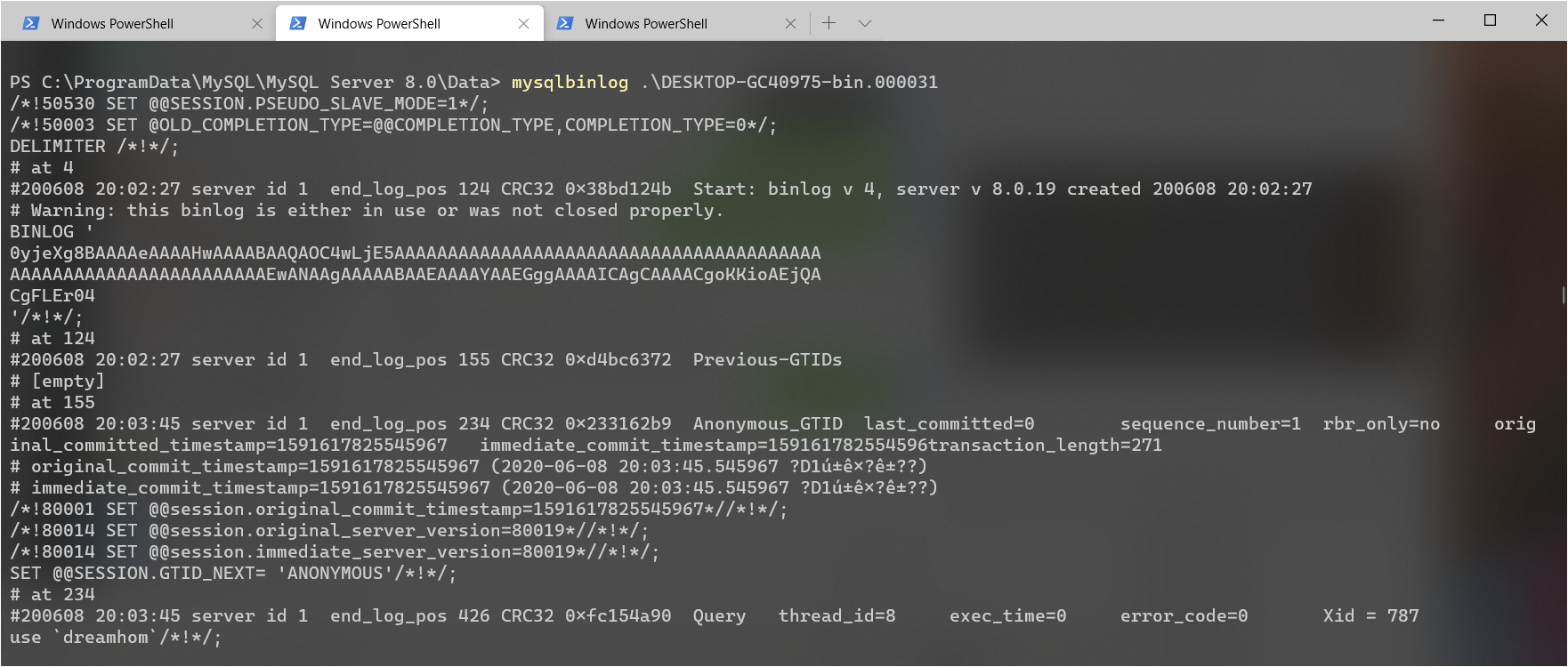
1. 逻辑备份：通过mysqldump进行全库备份与恢复
2. 逻辑备份：通过mysqldump+binlog进行备份与完整恢复
3. 逻辑备份：通过mysqldump+binlog进行增量备份与基于pos位置的恢复

1. **实验环境**

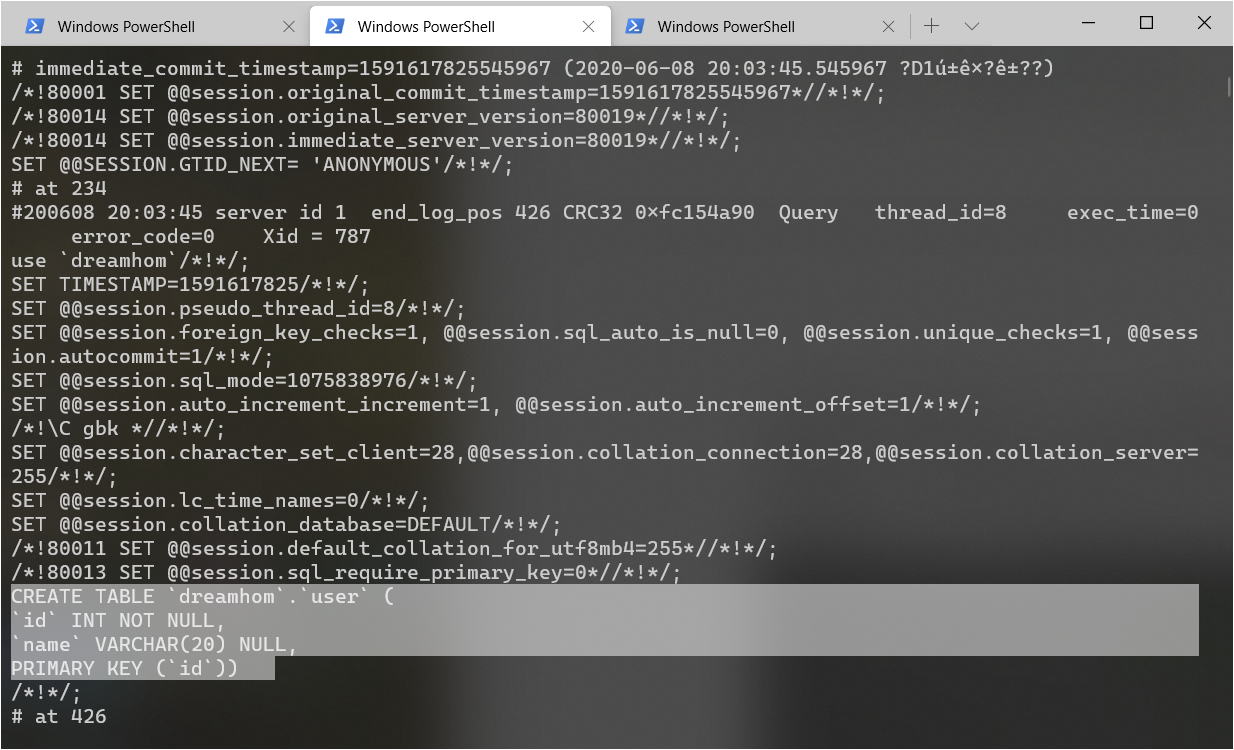
MySQL 8.0.19

1. **实验过程描述**
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）。

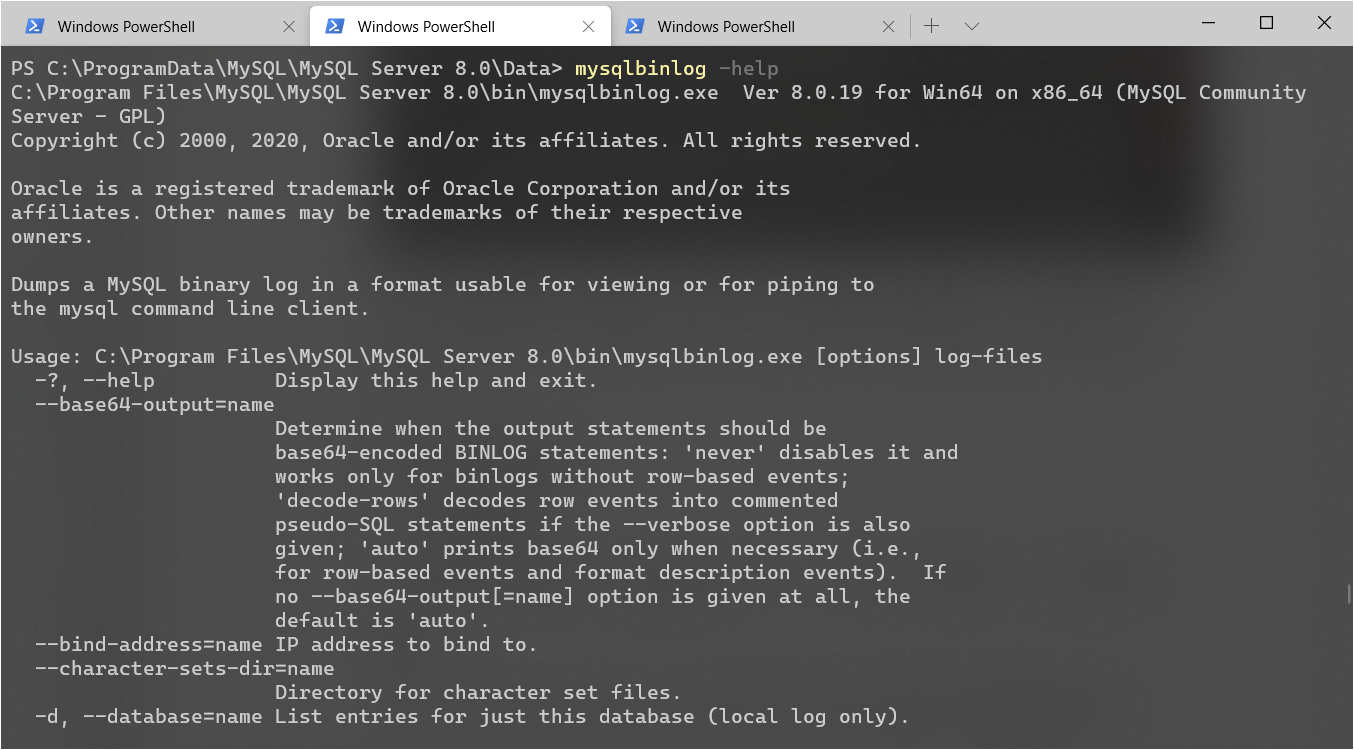
方法：使用mysqlbinlog命令+日志文件路径 打开日志文件，可以详细的看到经过的每一条操作的position（at和end\_log\_pos后面的数字分别代表开始和结束）以及所执行的SQL语句内容（除#开头外的语句），这样就可以通过查找SQL语句内容来获取日志的position。如图所示。



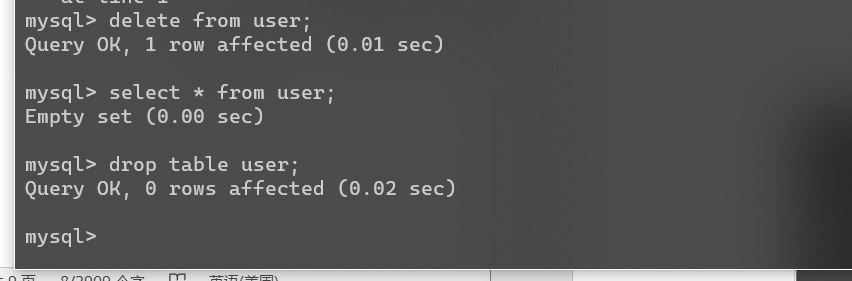
由下图内容可得知刚刚创建table‘user’的日志位于234-426

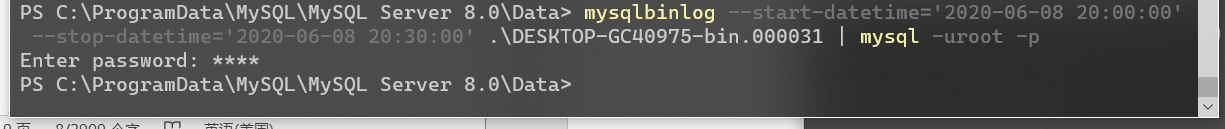


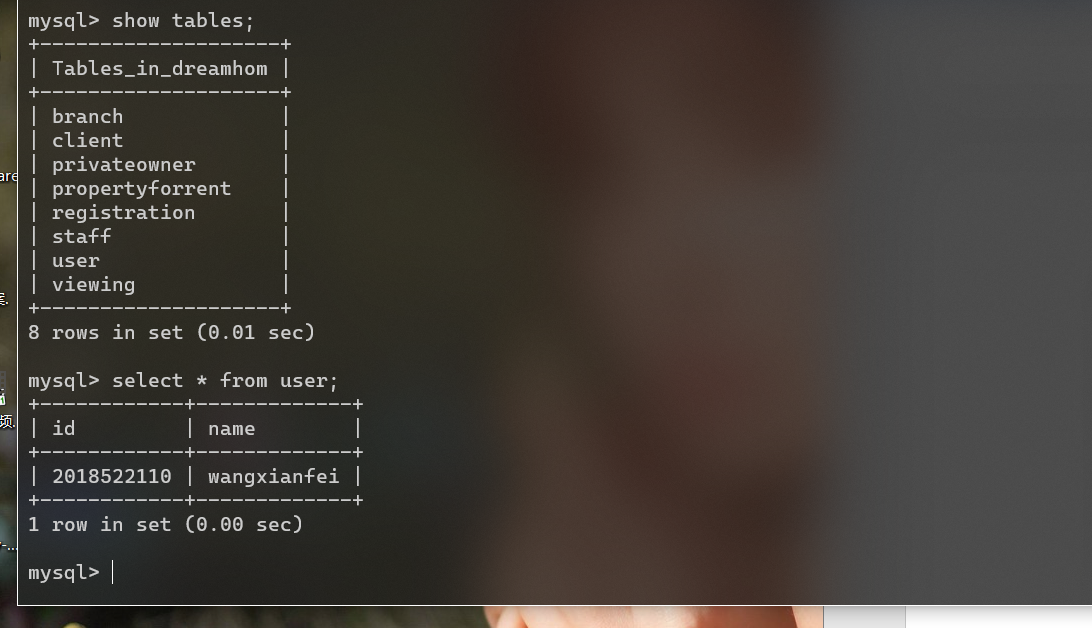
1. 除去以上的备份和恢复方式，在生产环境中，我们还时常面临基于时间进行数据恢复的状况，也就是恢复某一个时间段的数据，在上述实验的第三种备份和恢复策略中，稍作修改可以实现基于时间段的数据恢复，请问如何操作？给出详细的操作步骤（有图以及文字描述）。



方法：通过使用mysqlbinlog -help命令查看帮助文档（如上图）发现其可以指定--start-datetime 与 --stop-datetime参数，即可进行基于时间段的恢复操作。例如删除刚刚添加的user，然后基于时间恢复。（使用语句mysqlbinlog --start-datetime='2020-06-08 20:00:00' --stop-datetime='2020-06-08 20:30:00' .\DESKTOP-GC40975-bin.000031 | mysql -uroot -p）







1. **实验结果**

按照指导书中的步骤完成试验任务。并回答了指导书后提的两个问题。

心得：通过本次实验，学习到了如何对数据库进行恢复和备份，如何查看和使用数据库日志文件。