《数据库系统实验》

实验报告

|  |  |
| --- | --- |
| **题目** | 实验13 |
| **姓名** | 郝裕玮 |
| **学号** | 18329015 |
| **班级** | 计科1班 |

一、实验环境

MySQL 命令行 + MySQL Workbench 8.0 CE

二、实验内容与完成情况

针对教学管理系统（jxgl）做如下操作：

**12.1对表sc，course和student分别做表备份和表恢复。**

（1）表备份

在MySQL Workbench中执行以下语句：

use jxgl;

lock tables student read;

select \* into outfile 'student.bak' from student;

unlock tables;

lock tables sc read;

select \* into outfile 'sc.bak' from sc;

unlock tables;

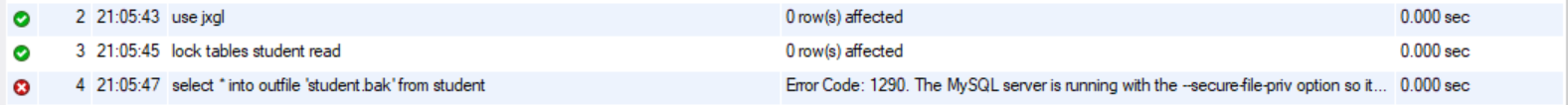
lock tables course read;

select \* into outfile 'course.bak' from course;

unlock tables;

然而在执行该行SQL语句时报错：

Error Code: 1290. The MySQL server is running with the --secure-file-priv option so it cannot execute this statement

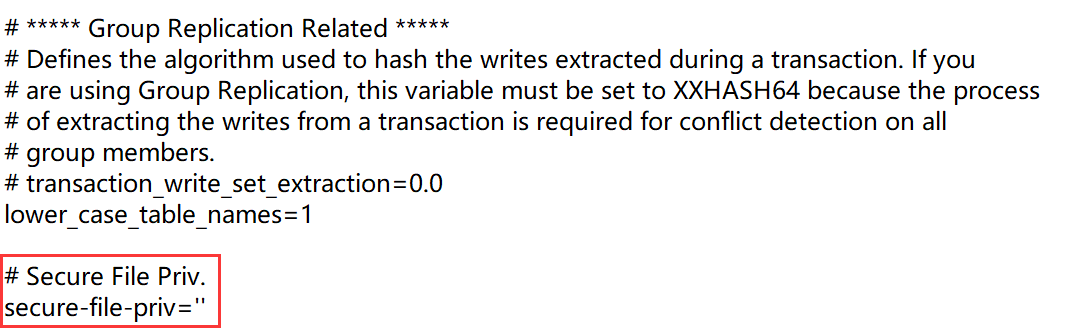


查询得知需要修改my.ini文件。

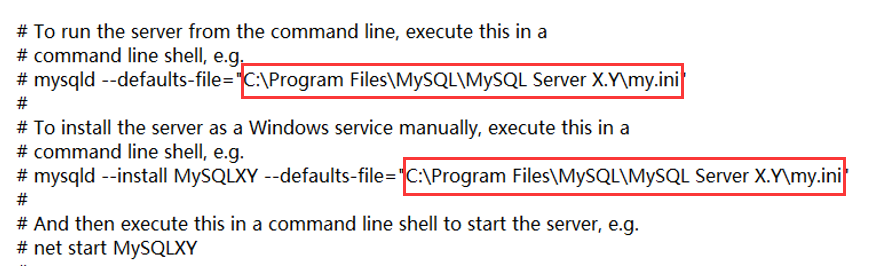
修改步骤如下：

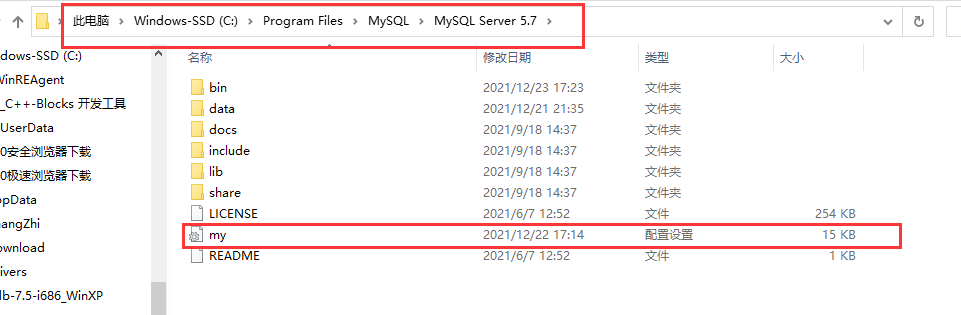
1. 文件夹中输入%ProgramData%并定位到C:\ProgramData\MySQL\MySQL Server 5.7

2. 修改my.ini文件，将secure-file-priv的定位目录修改为’’

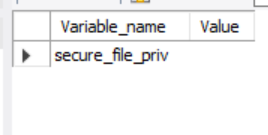


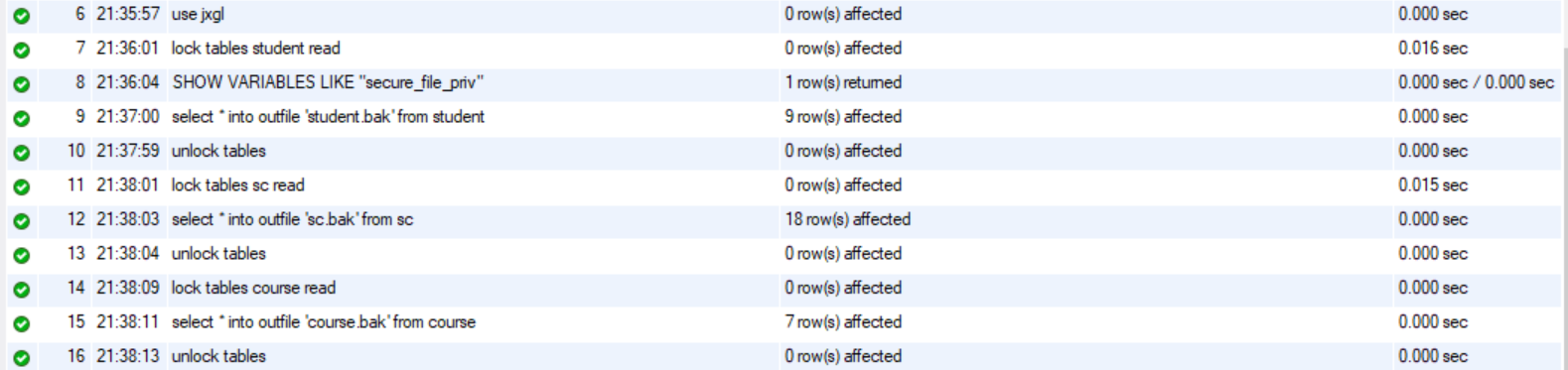
并根据下图，将my.ini文件复制到指定位置





复制完成后，打开cmd输入services.msc，找到MySQL服务，右键重启，并执行语句：SHOW VARIABLES LIKE "secure\_file\_priv"来查看secure-file-priv是否修改成功。若修改成功，则重新执行上述语句即可。





至此，备份成功！

（2）表恢复

在MySQL Workbench中执行以下语句：

set FOREIGN\_KEY\_CHECKS = 0;

lock tables student write;

load data infile 'student.bak' replace into table student;

unlock tables;

set FOREIGN\_KEY\_CHECKS = 1;

set FOREIGN\_KEY\_CHECKS = 0;

lock tables sc write;

load data infile 'sc.bak' replace into table sc;

unlock tables;

set FOREIGN\_KEY\_CHECKS = 1;

set FOREIGN\_KEY\_CHECKS = 0;

lock tables course write;

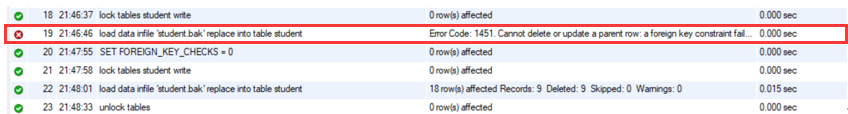
load data infile 'course.bak' replace into table course;

unlock tables;

set FOREIGN\_KEY\_CHECKS = 1;

一开始未加set FOREIGN\_KEY\_CHECKS = 0，直接执行第3行语句时会报错：

Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`jxgl`.`sc`, CONSTRAINT `sc\_ibfk\_1` FOREIGN KEY (`sno`) REFERENCES `student` (`sno`))



查询得知，该条记录的某个字段是作为innodb的另外一个表的外键。而在进行删除操作时会自动检查外键。

所以设置不在执行语句时不检查外键即可：

set FOREIGN\_KEY\_CHECKS = 0;

并在执行结束后将其重置为1

set FOREIGN\_KEY\_CHECKS = 1;

**12.2对教学管理系统采用不同方法进行完整备份和恢复。**

完整备份：

（1）MySQL脚本

以下3行语句均可备份：

mysqldump -u root -p jxgl > jxgl\_version1.sql

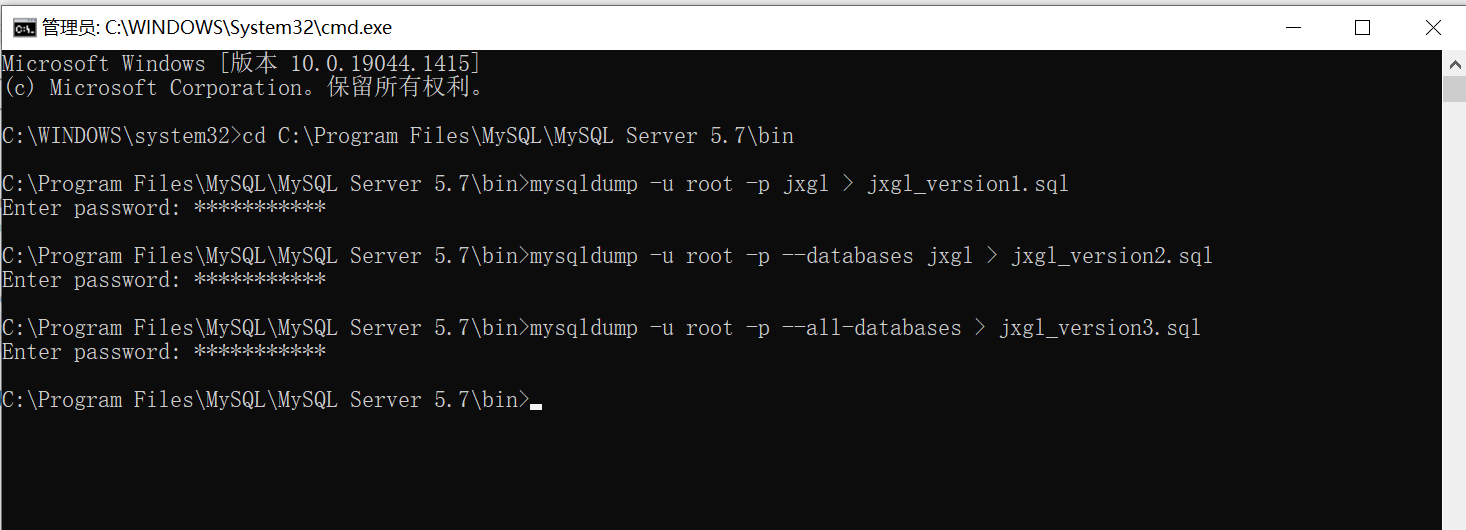
mysqldump -u root -p --databases jxgl > jxgl\_version2.sql

mysqldump -u root -p --all-databases > jxgl\_version3.sql

第1种：导出单个数据库

第2种：导出多个数据库

第3种：导出所有数据库

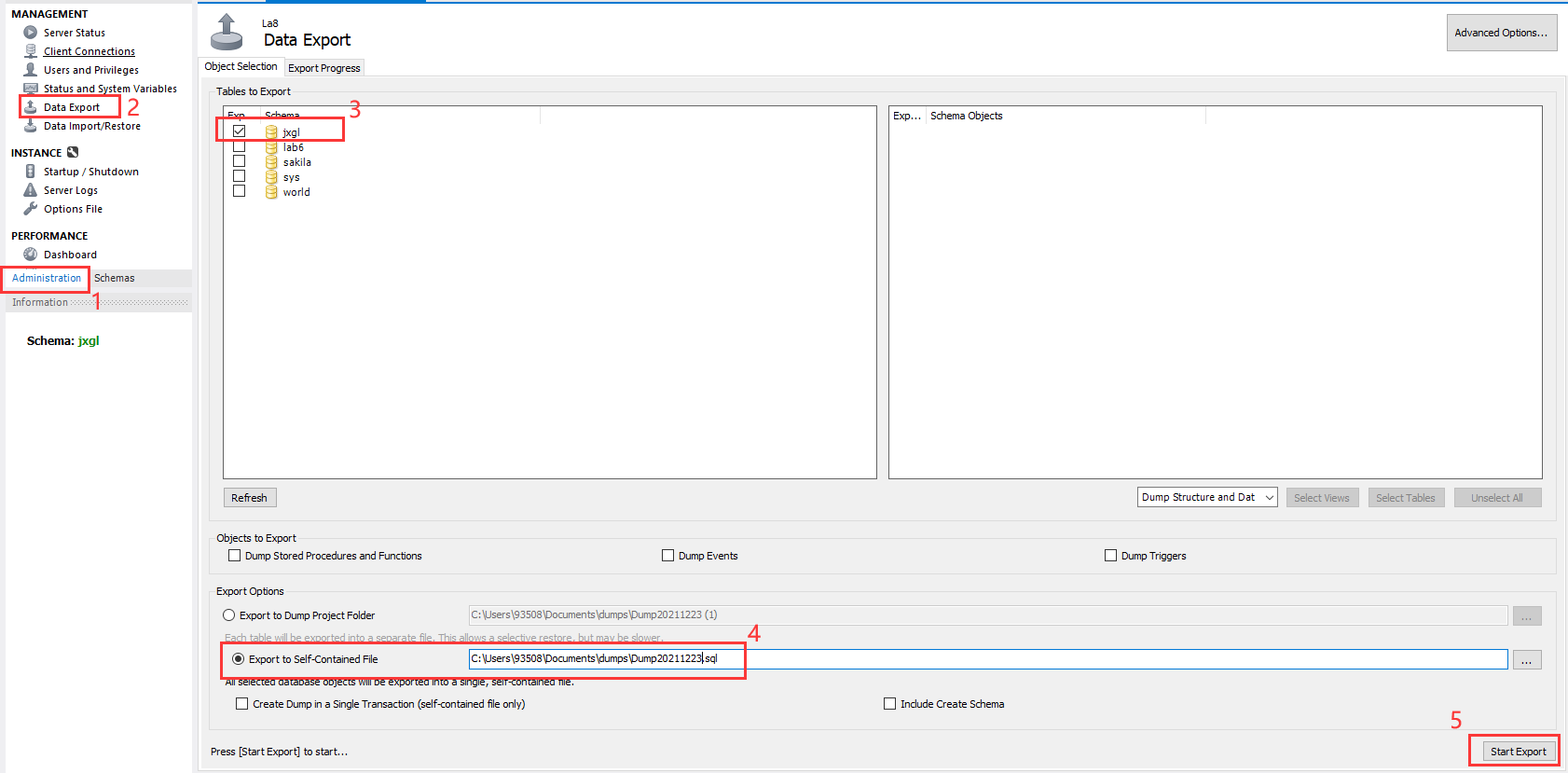


注意，以上语句需在cmd管理员模式下执行。

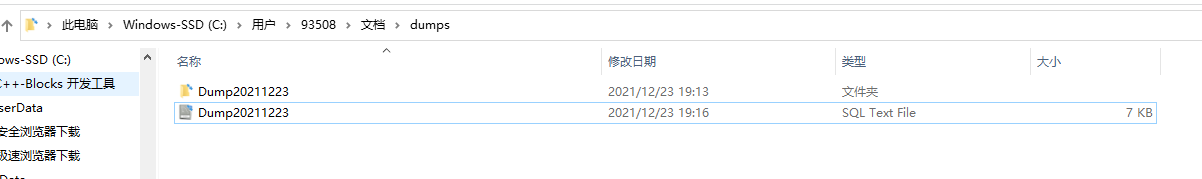
执行后可在对应文件夹下发现备份文件。



（2）MySQL Workbench图形化界面



执行后同样可在对应文件夹下发现备份文件。



完整恢复：

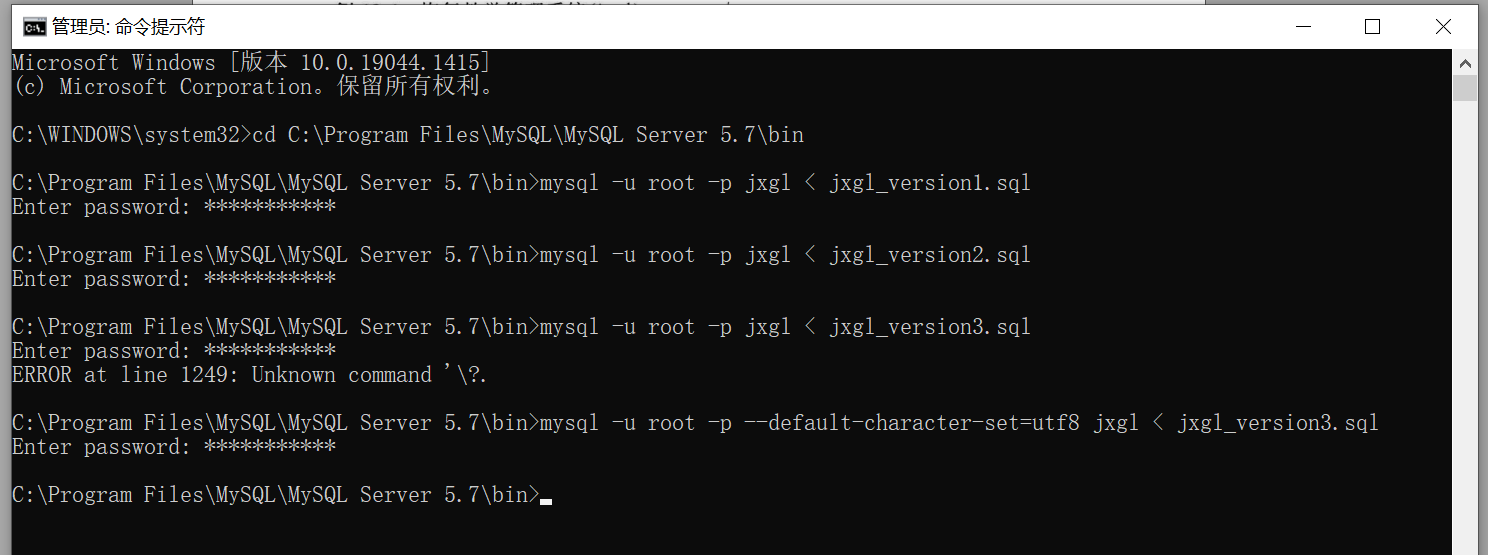
（1）MySQL脚本：

恢复语句为：

mysql -u root -p jxgl < jxgl\_version1.sql

mysql -u root -p jxgl < jxgl\_version2.sql

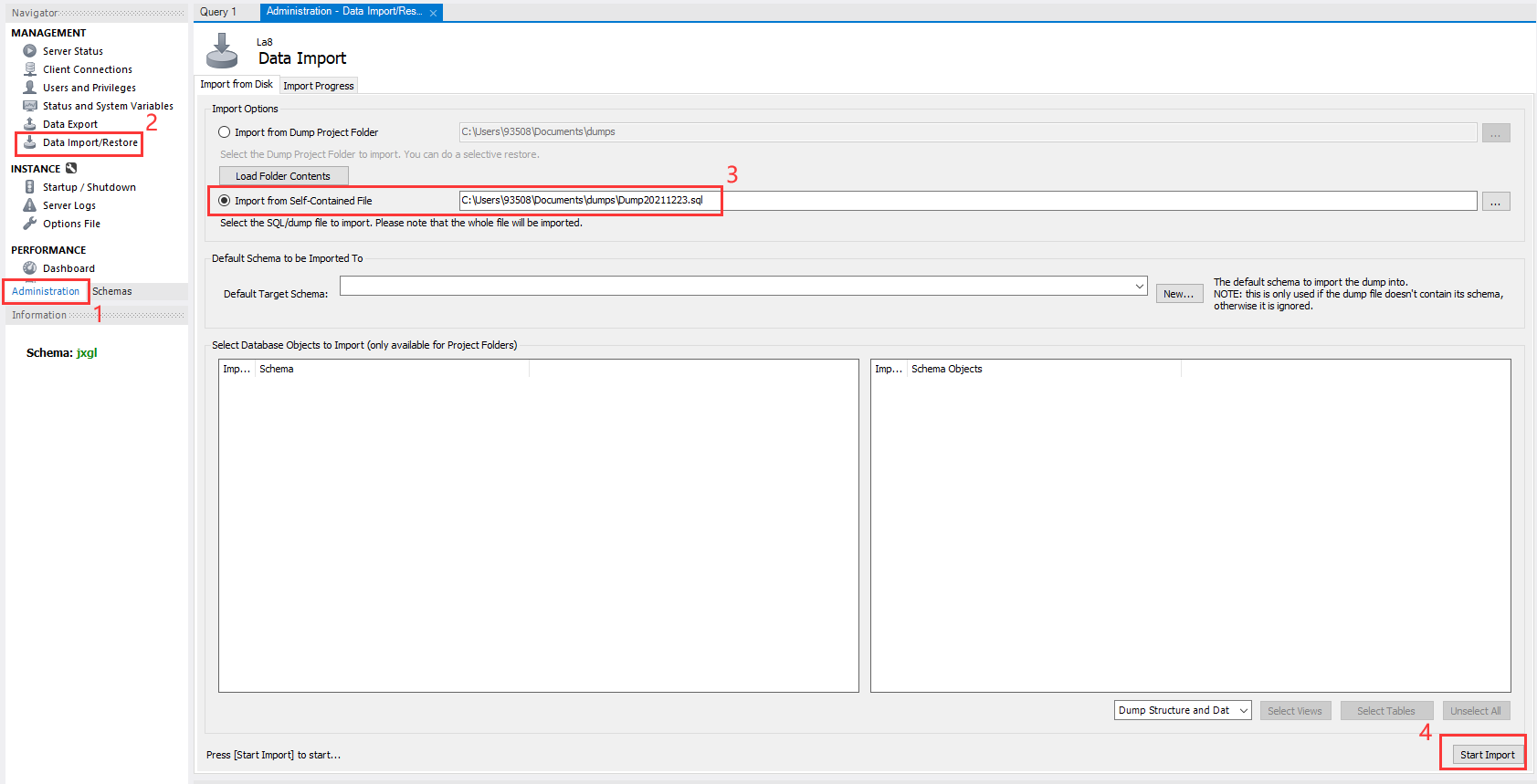
mysql -u root -p --default-character-set=utf8 jxgl < jxgl\_version3.sql



注意，以上语句需在cmd管理员模式下执行。

其中对于jxgl\_version3的备份恢复，由于字符集的问题，需要在导入命令中添加：--default-character-set=utf8

（2）MySQL Workbench图形化界面

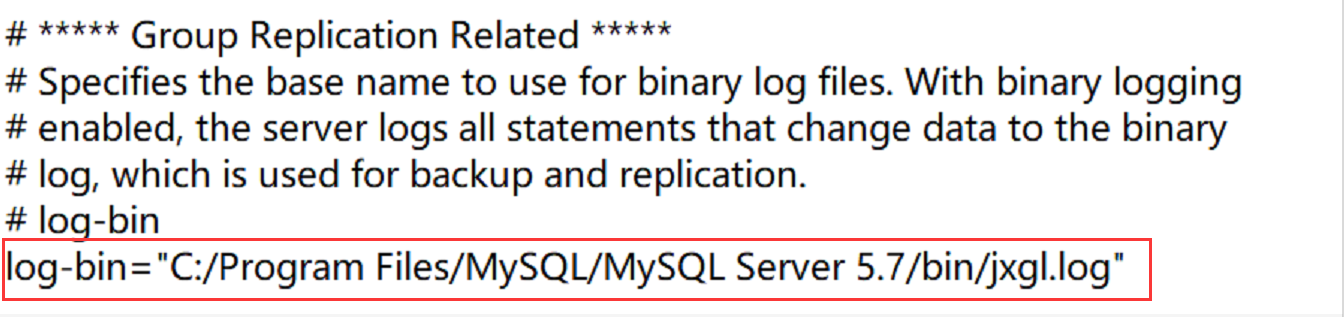


**12.3对教学管理系统与2011年6月29日上午9：00：00进行了差异备份；上午9：40数据库发生故障，根据其差异备份和日志文件进行位置恢复。**

首先需要再次修改my.ini文件，修改步骤如下：

1. 文件夹中输入%ProgramData%并定位到C:\ProgramData\MySQL\MySQL Server 5.7

2. 修改my.ini文件，设置log-bin的保存目录



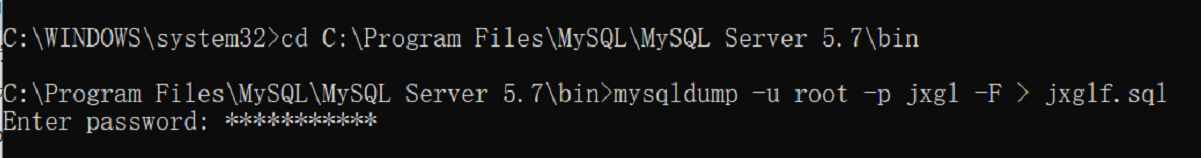
3. 将my.ini文件复制到之前的位置C:\Program Files\MySQL\MySQL Server 5.7

4. 复制完成后，打开cmd输入services.msc，找到MySQL服务，右键重启（重启后在log-bin对应的文件夹C:\Program Files\MySQL\MySQL Server 5.7\bin中会出现jxgl.index和jxgl.000001两个日志文件，可将其删除，便于后续操作）

之后的操作步骤如下：

（1）使用mysqldump -F（flush-logs）备份数据库

mysqldump -u root -p jxgl -F > jxglf.sql



注意，以上语句需在cmd管理员模式下执行。

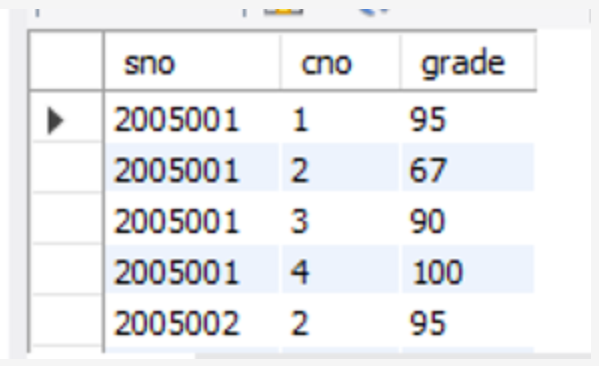
执行后发现log-bin对应的文件夹C:\Program Files\MySQL\MySQL Server 5.7\bin中再次出现jxgl.index和jxgl.000001两个日志文件。

（2）在MySQL Workbench中执行增删改（insert，delete，update）的操作，本次实验中执行操作如下：

insert into sc values('2005001','4',100);

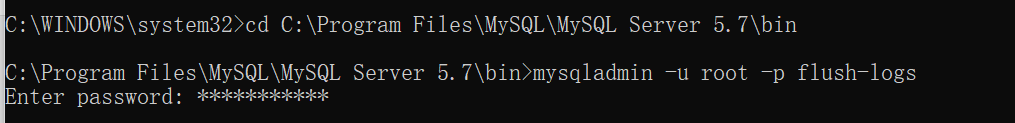
update sc set grade = 95 where sno = '2005001' and cno = '1';

执行后sc表变化如下：



（3）生成新的日志文件

mysqladmin -u root -p flush-logs

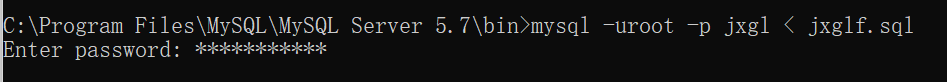


上述命令在cmd管理员模式下执行。

执行后会在log-bin对应的文件夹C:\Program Files\MySQL\MySQL Server 5.7\bin中发现有新的日志文件jxgl.000002

（4）执行mysql导入备份文件

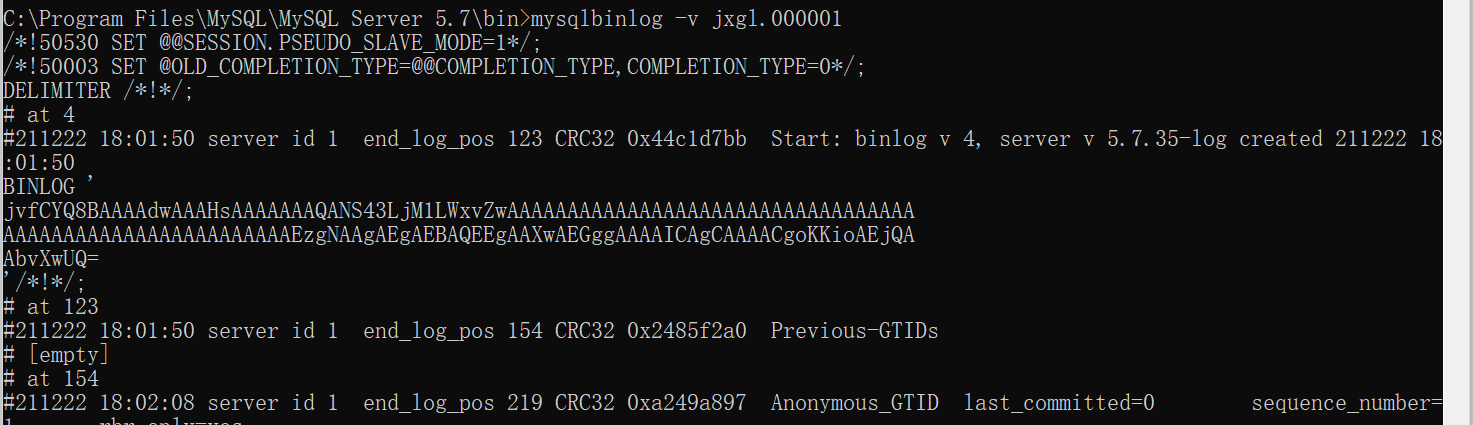
mysql -uroot -p jxgl < jxglf.sql



上述命令在cmd管理员模式下执行。

（5）查看日志文件

mysqlbinlog -v jxgl.000001

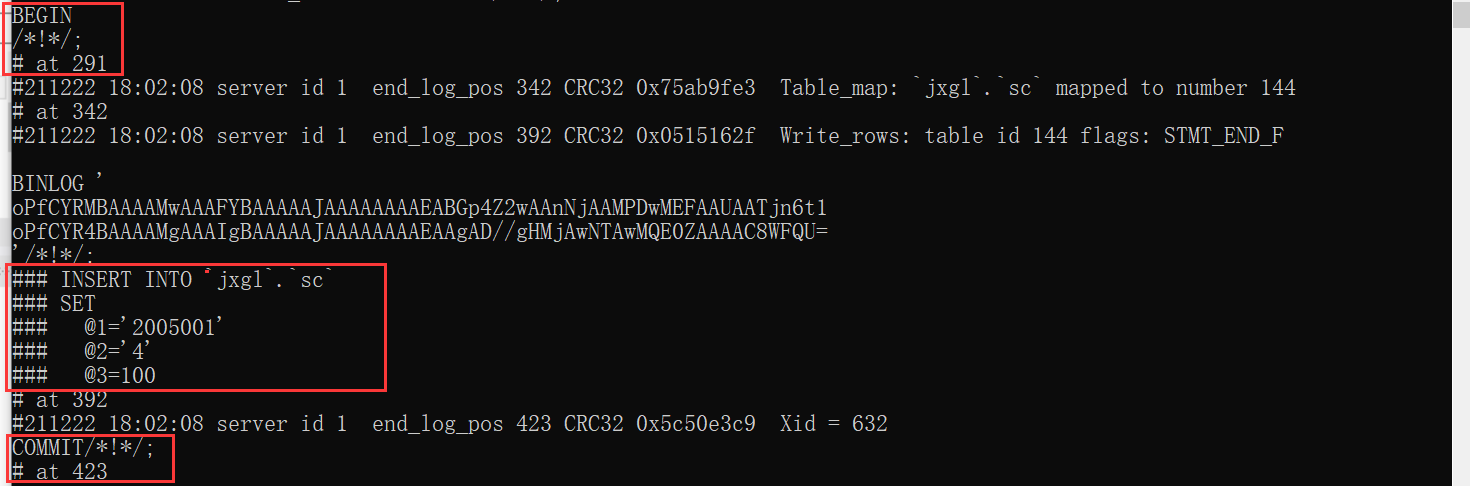


上述命令在cmd管理员模式下执行。

执行后在日志中寻找两个update语句执行前后的位置号，如下图所示：

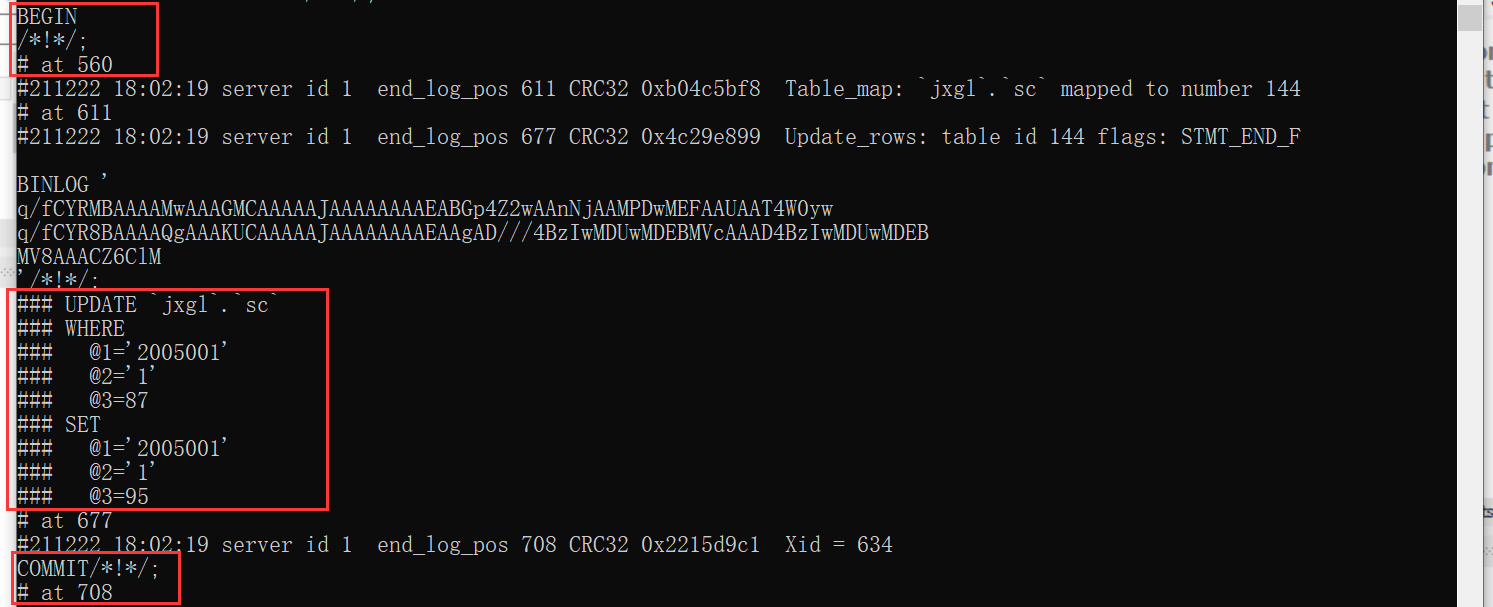
对于：

insert into sc values('2005001','4',100);



对于：

update sc set grade = 95 where sno = '2005001' and cno = '1';

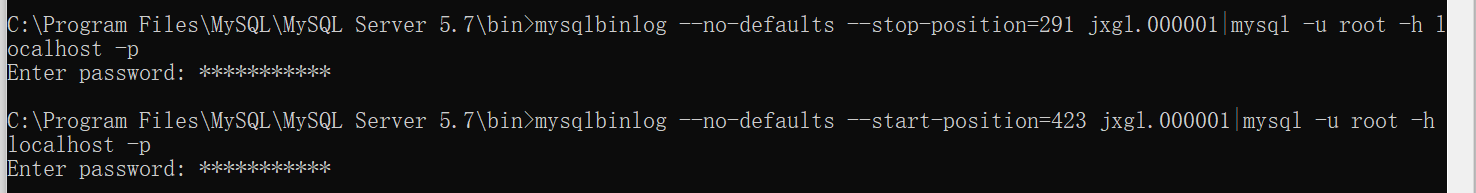


在这里我们选择跳过insert语句，但仍然执行update语句来进行位置恢复。

mysqlbinlog --no-defaults --stop-position=291 jxgl.000001|mysql -u root -h

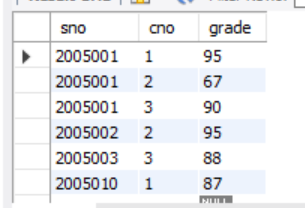
localhost -p

mysqlbinlog --no-defaults --start-position=423 jxgl.000001|mysql -u root -h localhost -p



上述命令在cmd管理员模式下执行。

执行后，在MySQL Workbench中查询sc：



发现成功跳过了insert语句，并继续执行update语句，位置恢复成功！