

数据库系统实验
实验报告

题目	(实验 13)
姓名	俞泽斌
学号	20337263
班级	计科 2 班

一、实验环境

- 1、操作系统：Windows 10
- 2、DBMS：mysql

二、实验内容

(1) 对表 sc, course 和 student 分别做表备份和表恢复

备份操作

先加锁后备份

```
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;
```

然后发现出现如下报错

```
ERROR 1290 (HY000): The MySQL server is running with the --secure-file-priv
option so it cannot execute this statement
```

查询资料，使用

```
show variables like '%secure%';
```

查看 secure-file-priv 当前的值是什么

```
mysql> show variables like '%secure%';
```

Variable_name	Value
require_secure_transport	OFF
secure_file_priv	C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\

```
2 rows in set (0.00 sec)
```

将路径添加到命令中，注意是/而非\

```
use jxgl;
lock tables student read;
select * into outfile 'C://ProgramData/MySQL/MySQL Server
8.0/Uploads/student.bak' from student;
unlock tables;
lock tables sc read;
select * into outfile 'C://ProgramData/MySQL/MySQL Server 8.0/Uploads/sc.bak'
from sc;
unlock tables;
lock tables course read;
select * into outfile 'C://ProgramData/MySQL/MySQL Server
8.0/Uploads/course.bak' from course;
unlock tables;
```

```
mysql> lock tables student read;
Query OK, 0 rows affected (0.00 sec)

mysql> select * into outfile 'C://ProgramData/MySQL/MySQL Server
8.0/Uploads/student.bak' from student;
Query OK, 8 rows affected (0.01 sec)

mysql> unlock tables;
Query OK, 0 rows affected (0.00 sec)

mysql> lock tables sc read;
Query OK, 0 rows affected (0.00 sec)

mysql> select * into outfile ' C://ProgramData/MySQL/MySQL Server
8.0/Uploads/sc.bak' from sc;
ERROR 1290 (HY000): The MySQL server is running with the --secure-file-priv
option so it cannot execute this statement
mysql> unlock tables;
Query OK, 0 rows affected (0.00 sec)

mysql> lock tables course read;
Query OK, 0 rows affected (0.00 sec)

mysql> select * into outfile 'C://ProgramData/MySQL/MySQL Server
8.0/Uploads/course.bak' from course;
Query OK, 7 rows affected (0.01 sec)

mysql> unlock tables;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> lock tables sc read;
Query OK, 0 rows affected (0.00 sec)

mysql> select * into outfile 'C://ProgramData/MySQL/MySQL Server
8.0/uploads/sc.bak' from sc;
Query OK, 18 rows affected (0.01 sec)

mysql> unlock tables;
Query OK, 0 rows affected (0.00 sec)
```

结果

```
mysql> lock tables student read;
Query OK, 0 rows affected (0.00 sec)

mysql> select * into outfile 'C://ProgramData/MySQL/MySQL Server
8.0/uploads/student.bak' from student;
Query OK, 8 rows affected (0.01 sec)

mysql> unlock tables;
Query OK, 0 rows affected (0.00 sec)

mysql> lock tables sc read;
Query OK, 0 rows affected (0.00 sec)

mysql> select * into outfile 'C://ProgramData/MySQL/MySQL Server
8.0/uploads/sc.bak' from sc;
ERROR 1290 (HY000): The MySQL server is running with the --secure-file-priv
option so it cannot execute this statement
mysql> unlock tables;
Query OK, 0 rows affected (0.00 sec)

mysql> lock tables course read;
Query OK, 0 rows affected (0.00 sec)

mysql> select * into outfile 'C://ProgramData/MySQL/MySQL Server
8.0/uploads/course.bak' from course;
Query OK, 7 rows affected (0.01 sec)

mysql> unlock tables;
Query OK, 0 rows affected (0.00 sec)

mysql> lock tables sc read;
Query OK, 0 rows affected (0.00 sec)

mysql> select * into outfile 'C://ProgramData/MySQL/MySQL Server
8.0/uploads/sc.bak' from sc;
Query OK, 18 rows affected (0.01 sec)

mysql> unlock tables;
Query OK, 0 rows affected (0.00 sec)
```

完成备份

恢复

首先我们看一下最初的sc表

```
select * from sc;
```

结果

```
+-----+-----+-----+
| sno    | cno  | grade |
+-----+-----+-----+
| 2005001 | 1    | 87    |
| 2005002 | 2    | 94    |
| 2005002 | 3    | NULL  |
| 2005003 | 3    | 88    |
| 2005003 | 7    | NULL  |
| 2005010 | 2    | NULL  |
| 2005010 | 3    | 80    |
| 2005010 | 4    | 87    |
| 2005010 | 6    | 85    |
| 2005011 | 1    | NULL  |
| 2005011 | 2    | 48    |
| 2005011 | 3    | 56    |
| 2005011 | 5    | 47    |
| 2005012 | 1    | 84    |
| 2005012 | 3    | NULL  |
| 2005012 | 4    | 67    |
| 2005012 | 5    | 81    |
| 2005012 | 7    | NULL  |
+-----+-----+-----+
18 rows in set (0.00 sec)
```

做以下改动

```
update sc set grade=99 where sno='2005001';
```

```
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

查看

```
select * from sc where sno='2005001';
```

```
+-----+-----+-----+
| sno    | cno  | grade |
+-----+-----+-----+
| 2005001 | 1    | 99    |
+-----+-----+-----+
1 row in set (0.00 sec)
```

之后进行恢复

```
SET FOREIGN_KEY_CHECKS = 0;
lock tables student write;
load data infile 'C://ProgramData/MySQL/MySQL Server 8.0/Uploads/student.bak'
replace into table student;
unlock tables;
set foreign_key_checks = 1;
lock tables sc write;
load data infile 'C://ProgramData/MySQL/MySQL Server 8.0/Uploads/sc.bak' replace
into table sc;
unlock tables;
SET FOREIGN_KEY_CHECKS = 0;
lock tables course write;
load data infile 'C://ProgramData/MySQL/MySQL Server 8.0/Uploads/course.bak'
replace into table course;
unlock tables;
set foreign_key_checks = 1;
```

运行结果

```
mysql> SET FOREIGN_KEY_CHECKS = 0;
Query OK, 0 rows affected (0.00 sec)

mysql> lock tables student write;
Query OK, 0 rows affected (0.00 sec)

mysql> load data infile 'C://ProgramData/MySQL/MySQL Server
8.0/Uploads/student.bak' replace into table student;
Query OK, 8 rows affected (0.01 sec)
Records: 8 Deleted: 0 Skipped: 0 Warnings: 0

mysql> unlock tables;
Query OK, 0 rows affected (0.00 sec)

mysql> set foreign_key_checks = 1;
Query OK, 0 rows affected (0.00 sec)

mysql> lock tables sc write;
Query OK, 0 rows affected (0.00 sec)

mysql> load data infile 'C://ProgramData/MySQL/MySQL Server 8.0/Uploads/sc.bak'
replace into table sc;
Query OK, 19 rows affected (0.00 sec)
Records: 18 Deleted: 1 Skipped: 0 Warnings: 0

mysql> unlock tables;
Query OK, 0 rows affected (0.00 sec)

mysql> SET FOREIGN_KEY_CHECKS = 0;
Query OK, 0 rows affected (0.00 sec)

mysql> lock tables course write;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> load data infile 'C://ProgramData/MySQL/MySQL Server
8.0/Uploads/course.bak' replace into table course;
Query OK, 14 rows affected (0.00 sec)
Records: 7 Deleted: 7 Skipped: 0 Warnings: 0

mysql> unlock tables;
Query OK, 0 rows affected (0.00 sec)

mysql> set foreign_key_checks = 1;
Query OK, 0 rows affected (0.00 sec)
```

此时再次查看sc表中我们修改的部分

```
select * from sc where sno='2005001';
```

运行结果

```
mysql> select * from sc where sno='2005001';
+-----+-----+-----+
| sno   | cno | grade |
+-----+-----+-----+
| 2005001 | 1   | 87    |
+-----+-----+-----+
1 row in set (0.00 sec)
```


可以看到此时也恢复了改之前的数据，完成了恢复

(2) 对教学管理系统采用不同方法进行完整备份和恢复

方法一：

```
mysqldump -uroot -p jxgl>jxgl.sql
```

输入密码后生成备份文件

 jxgl.sql	2022/12/14 18:07	SQL Text File	5,630 KB
--	------------------	---------------	----------

重新对sc表进行修改来测试

```
use jxgl;
update sc set grade=0 where sno='2005001';
select * from sc where sno='2005001';
```

结果为

```
mysql> update sc set grade=0 where sno='2005001';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
mysql> select * from sc where sno='2005001';
+-----+-----+-----+
| sno    | cno  | grade |
+-----+-----+-----+
| 2005001 | 1    | 0     |
+-----+-----+-----+
1 row in set (0.00 sec)
```

此时我们将备份恢复

```
mysql -uroot -p jxgl<jxgl.sql
```

```
C:\Users\AholiC\y>mysql -uroot -p jxgl<jxgl.sql
Enter password: *****
```



重新选择上面改动后的数据

```
mysql> select * from sc where sno='2005001';
+-----+-----+-----+
| sno    | cno  | grade |
+-----+-----+-----+
| 2005001 | 1    | 87     |
+-----+-----+-----+
1 row in set (0.00 sec)
```

可以看到得到的是改动前的数据

方法二

```
mysqldump -uroot -p --databases jxgl>jxgl2.sql
```

 jxgl.sql	2022/12/14 18:07	SQL Text File	5,630 KB
 jxgl2.sql	2022/12/14 18:18	SQL Text File	5,630 KB

也做上述的修改操作

```
use jxgl;
update sc set grade=0 where sno='2005001';
select * from sc where sno='2005001';
```

结果

```
mysql> use jxgl;
Database changed
mysql> update sc set grade=0 where sno='2005001';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from sc where sno='2005001';
+-----+-----+-----+
| sno    | cno  | grade |
+-----+-----+-----+
| 2005001 | 1    | 0     |
+-----+-----+-----+
1 row in set (0.00 sec)
```

然后恢复

```
mysql -uroot -p jxgl<jxgl2.sql
```

```
C:\Users\AholiC^y>mysql -uroot -p jxgl<jxgl2.sql
Enter password: *****
```

重新选择上面改动后的数据




```
mysql> use jxgl;
Database changed
mysql> select * from sc where sno='2005001';
+-----+-----+-----+
| sno    | cno  | grade |
+-----+-----+-----+
| 2005001 | 1    | 87    |
+-----+-----+-----+
1 row in set (0.00 sec)
```

恢复成功

方法三:

```
mysqldump -uroot -p --all-databases>jxgl3.sql
```

```
C:\Users\AholiC^y>mysqldump -uroot -p --all-databases>jxgl3.sql
Enter password: *****
```

 jxgl.sql	2022/12/14 18:07	SQL Text File	5,630 KB
 jxgl2.sql	2022/12/14 18:18	SQL Text File	5,630 KB
 jxgl3.sql	2022/12/14 18:22	SQL Text File	10,436 KB

重复上面的修改


```
mysql> use jxgl;
Database changed
mysql> update sc set grade=0 where sno='2005001';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from sc where sno='2005001';
+-----+-----+-----+
| sno    | cno  | grade |
+-----+-----+-----+
| 2005001 | 1    | 0     |
+-----+-----+-----+
1 row in set (0.00 sec)
```

恢复

```
mysql -uroot -p jxgl<jxgl3.sql
```

```
C:\Users\AholiC^y>mysql -uroot -p jxgl<jxgl3.sql
Enter password: *****
```

重新选择上面改动后的数据

```
mysql> use jxgl;
Database changed
mysql> select * from sc where sno='2005001';
+-----+-----+-----+
| sno    | cno  | grade |
+-----+-----+-----+
| 2005001 | 1    | 87    |
+-----+-----+-----+
1 row in set (0.00 sec)
```

恢复成功

(3) 对教学管理系统与 2022 年 12 月 15日上午 12:12 进行了差异备份； 上午 12: 15 数据库发生故障， 根据其差异备份和日志文件进行时间点恢复。

差异备份恢复

首先查看日志是否已经启用

```
show variables like '%log_bin%';
```

```
mysql> show variables like '%log_bin%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| log_bin       | ON   |
| log_bin_basename | C:\ProgramData\MySQL\MySQL Server 8.0\Data\LAPTOP-BH85TEHU-bin |
| log_bin_index  | C:\ProgramData\MySQL\MySQL Server 8.0\Data\LAPTOP-BH85TEHU-bin.index |
| log_bin_trust_function_creators | OFF |
| log_bin_use_v1_row_events | OFF |
| sql_log_bin    | ON   |
+-----+-----+
6 rows in set (0.01 sec)
```

可以看到此时的日志是启用状态

那么开始进行差异备份

```
mysqldump -uroot -p jxgl -F>jxglf.sql
```

同时备份日志文件binlog

```
mysqladmin -uroot -p flush-logs
```

名称	创建时间	权限设置	大小
jxgl.sql	2022/12/14 18:07	SQL Text File	5,630 KB
jxgl2.sql	2022/12/14 18:18	SQL Text File	5,630 KB
jxgl3.sql	2022/12/14 18:22	SQL Text File	10,436 KB
jxglf.sql	2022/12/15 12:14	SQL Text File	5,630 KB

开始对数据库进行修改

```
use jxgl;  
update sc set grade=66 where sno='2005001';  
select * from sc where sno='2005001';
```

```
mysql> use jxgl;  
Database changed  
mysql> update sc set grade=66 where sno='2005001';  
Query OK, 1 row affected (0.02 sec)  
Rows matched: 1 Changed: 1 Warnings: 0  
  
mysql> select * from sc where sno='2005001';  
+-----+-----+-----+  
| sno   | cno | grade |  
+-----+-----+-----+  
| 2005001 | 1   | 66    |  
+-----+-----+-----+  
1 row in set (0.00 sec)
```

然后用备份差异恢复

```
mysql -u root -p jxgl<jxglf.sql
```

重新查询

```
mysql> use jxgl;  
Database changed  
mysql> select * from sc where sno='2005001';  
+-----+-----+-----+  
| sno   | cno | grade |  
+-----+-----+-----+  
| 2005001 | 1   | 87    |  
+-----+-----+-----+  
1 row in set (0.00 sec)
```

恢复到之前版本

使用日志文件进行恢复操作

还是进行上述的修改操作

```
mysql> use jxgl;
Database changed
mysql> update sc set grade=66 where sno='2005001';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from sc where sno='2005001';
+-----+-----+-----+
| sno    | cno  | grade |
+-----+-----+-----+
| 2005001 | 1    | 66    |
+-----+-----+-----+
1 row in set (0.00 sec)
```

然后我们查看一下现在的bin文件名

```
show master status;
```

```
+-----+-----+-----+-----+
-----+
| File                               | Position | Binlog_Do_DB | Binlog_Ignore_DB |
Executed_Gtid_Set |
+-----+-----+-----+-----+
-----+
| LAPTOP-BH85TEHU-bin.000023 | 3595738 |              |                  |
|                               |         |              |                  |
+-----+-----+-----+-----+
-----+
1 row in set (0.00 sec)
```

此时我们可以通过两种方式查看具体的位置信息

第一种直接在控制台通过mysql查看

```
show binlog events in 'LAPTOP-BH85TEHU-bin.000023';
```

```
| LAPTOP-BH8STEHU-bin.000023 | 3595189 | Xid | 1 | 3595220 | COMMIT /* xid=3065 */  
| LAPTOP-BH8STEHU-bin.000023 | 3595220 | Anonymous_Gtid | 1 | 3595297 | SET @@SESSION.GTID_NEXT= 'ANONYMOUS'  
| LAPTOP-BH8STEHU-bin.000023 | 3595297 | Query | 1 | 3595425 | use `jxgl`; /*!40000 ALTER TABLE `ts` ENABLE KEYS */ /* xid  
=3066 */  
| LAPTOP-BH8STEHU-bin.000023 | 3595425 | Anonymous_Gtid | 1 | 3595504 | SET @@SESSION.GTID_NEXT= 'ANONYMOUS'  
| LAPTOP-BH8STEHU-bin.000023 | 3595504 | Query | 1 | 3595588 | BEGIN  
| LAPTOP-BH8STEHU-bin.000023 | 3595588 | Table_map | 1 | 3595645 | table_id: 361 (jxgl.sc)  
| LAPTOP-BH8STEHU-bin.000023 | 3595645 | Update_rows | 1 | 3595707 | table_id: 361 flags: STMT_END_F  
| LAPTOP-BH8STEHU-bin.000023 | 3595707 | Xid | 1 | 3595738 | COMMIT /* xid=3081 */  
  
+-----+-----+-----+-----+-----+  
+-----+-----+-----+-----+-----+  
+-----+-----+-----+-----+-----+  
538 rows in set (0.01 sec)
```

只截取了后面一部分，也可以看到最后一条语句执行的位置

或者采用mysqlbinlog编译后用记事本查看

输入命令

```
mysqlbinlog LAPTOP-BH85TEHU-bin.000023> jxqltemp.sql
```

```

SET @@SESSION.foreign_key_checks=1, @@session.unique_checks=1/*!*/;
SET @@session.sql_mode=1075838976/*!*/;
BEGIN
/*!*/;
# at 3595504
#221215 12:29:22 server id 1  end_log_pos 3595588 CRC32 0x85ba45ee      Query   thread_id=33      exec_time=0      error_code=0
SET TIMESTAMP=1671078562/*!*/;
SET @@session.foreign_key_checks=1, @@session.unique_checks=1/*!*/;
SET @@session.sql_mode=1075838976/*!*/;
BEGIN
/*!*/;
# at 3595588
#221215 12:29:22 server id 1  end_log_pos 3595645 CRC32 0x305a53a3      Table_map: `jxgl`.`sc` mapped to number 361
# at 3595645
#221215 12:29:22 server id 1  end_log_pos 3595707 CRC32 0x9724bf6d      Update_rows: table id 361 flags: STMT_END_F

BINLOG '
oqKaYxMBAAAAOQAAAH3dNgAAAGkBAAAAAAEABGp4Z2wAAANjAAP+/gIE/hX+BgQBAQACASGjU1ow
oqKaYx8BAAAAAPgAAALvdNgAAAGkBAAAAAAEAAgAD//8ABzlwMDUwMDEBMVCAAcyMDA1MDAxATFC
AG2/JJc=
'/*!*/;
# at 3595707
#221215 12:29:22 server id 1  end_log_pos 3595738 CRC32 0x915d2b42      Xid = 3081
COMMIT/*!*/;
SET @@SESSION.GTID_NEXT= 'AUTOMATIC' /* added by mysqlbinlog *//*!*/;
DELIMITER ;
# End of log file
/*!50003 SET COMPLETION_TYPE=@OLD_COMPLETION_TYPE*/;
/*!500530 SET @@SESSION.PSEUDO_SLAVE_MODE=0*/;

```

查看文件可以看到update在3595588的位置，最后一条语句在3595707位置上，所以输入命令

```
mysqlbinlog --no-defaults --stop-position=3595588 LAPTOP-BH85TEHU-  
bin.000023|mysql -u root -h localhost -p  
mysqlbinlog --no-defaults --start-position=3595707 LAPTOP-BH85TEHU-  
bin.000023|mysql -u root -h localhost -p
```

```
C:\ProgramData\MySQL\MySQL Server 8.0\Data>mysqlbinlog --no-defaults --stop-position=3595588 LAPTOP-BH85TEHU-bin.000023
mysql -u root -h localhost -p
Enter password: *****

C:\ProgramData\MySQL\MySQL Server 8.0\Data>mysqlbinlog --no-defaults --start-position=3595707 LAPTOP-BH85TEHU-bin.000023
mysql -u root -h localhost -p
Enter password: *****
```

实现恢复

验证

```
mysql> use jxgl;  
Database changed  
mysql> select * from sc where sno='2005001';  
+-----+-----+-----+  
| sno    | cno | grade |  
+-----+-----+-----+  
| 2005001 | 1   | 87    |  
+-----+-----+-----+  
1 row in set (0.00 sec)
```

可以看到此时完成了位置恢复