《数据库系统实验》

实验报告

题目	(实验 13)
姓名	TRY
学号	
班级	计科 X 班

一. 实验环境:

操作系统: windows

图形界面: mysql3.7.31, mysql workbench

二. 实验内容与完成情况:

本次实验在以前建立的教学管理系统(jxgl)的基础上完成。

2.1 对表 sc, course 和 student 分别做表备份和表恢复

2.1.1 备份表

代码:

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;

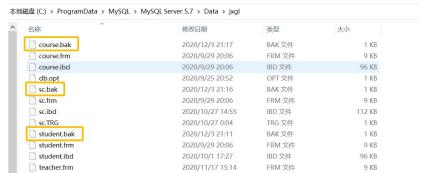
lock tables course read;

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

unlock tables;

结果:

	#		Time	Action	Message
0		1	21:11:42	lock tables student read	0 row(s) affected
0		2	21:11:44	select * into outfile 'student.bak' from student	10 row(s) affected
0		3	21:16:17	unlock tables	0 row(s) affected
0		4	21:16:47	lock tables sc read	0 row(s) affected
0		5	21:16:47	select * into outfile 'sc.bak' from sc	29 row(s) affected
0		6	21:16:47	unlock tables	0 row(s) affected
0		7	21:17:05	lock tables course read	0 row(s) affected
0		8	21:17:05	select * into outfile 'course.bak' from course	7 row(s) affected
0		9	21:17:05	unlock tables	0 row(s) affected



生成了3个文件。

报错:

2 20:59:51 select * into outfile 'student bak' from student Error Code: 1290. The MySQL server is running with the --secure-file-priv option so it cannot execute this statement

在执行第二行的时候,报了上述的错误。

解决办法:



在 my. ini 文件中,修改 secure_file_priv 变量为 secure_file_priv='', 并重启 mysql 服务器即可。

2.1.2 表恢复

代码:

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;

lock tables sc write;

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

unlock tables;

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;

结果:

0	1	4 21:30:24	SET FOREIGN_KEY_CHECKS = 0	0 row(s) affected	0.000 sec
0	1	5 21:30:24	lock tables student write	0 row(s) affected	0.000 sec
0	1	6 21:30:24	load data infile "student.bak" replace into table student	20 row(s) affected Records: 10 Deleted: 10 Skipped: 0 Warnings: 0	0.031 sec
0	1	7 21:30:24	unlock tables	0 row(s) affected	0.000 sec
0	1	8 23:14:09	set foreign_key_checks = 1	0 row(s) affected	0.015 sec
0	1	9 23:14:09	lock tables sc write	0 row(s) affected	0.015 sec
0	2	0 23:14:09	load data infile 'sc.bak' replace into table sc	29 row(s) affected Records: 29 Deleted: 0 Skipped: 0 Warnings: 0	0.047 sec
0	2	1 23:14:09	unlock tables	0 row(s) affected	0.000 sec

(•	25	23:15:08	SET FOREIGN_KEY_CHECKS = 0	0 row(s) affected	0.000 sec
(0	26	23:15:08	lock tables course write	0 row(s) affected	0.000 sec
(0	27	23:15:08	load data infile 'course.bak' replace into table course	14 row(s) affected Records: 7 Deleted: 7 Skipped: 0 Warnings: 0	0.047 sec
	0	28	23:15:08	unlock tables	0 row(s) affected	0.000 sec
	0	29	23:15:08	set foreign_key_checks = 1	0 row(s) affected	0.000 sec

报错:

2 12 21:23:31 load data infile 'student.bak' replace into table student

Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails ('jxgl'.'sc', CONSTRAIN...

解决办法:

这是外码错误,在执行语句前加入语句"SET FOREIGN_KEY_CHECKS = 0;"即可。

2.2 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

C:\Users\lenovo>mysqldump -uroot -p jxgl>jxgl.sql Enter password: ***********

ixgl

2020/12/3 23:38

SQL Text File

5,305 KB

方法二: mysqldump -uroot -p --databases jxgl>jxgl2.sql

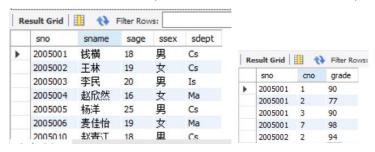
E jxgl2 2020/12/3 23:51 SQL Text File 5,305 KB

方法三: mysqldump -uroot -p --all-databases>jxgl3.sql

i jxgl3 2020/12/4 0:06 SQL Text File 9,729 KB

修改数据库:

update sc set grade = 90 where cno = 1 and sno = "2005001"; insert into student values("2005006","麦佳怡",19,"女","Ma");



还原数据库:

mysql -uroot -p jxgl<jxgl.sql

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

mysql -uroot -p jxgl<jxgl2.sql

C:\Users\lenovo>mysql -uroot -p jxgl<jxgl2.sql Enter password: ************

mysql -uroot -p jxgl<jxgl3.sql

C:\Users\lenovo>mysql -uroot -p jxgl<jxgl3.sql Enter password: *************

恢复结果:经过验证,三种方法备份都能如下面一样恢复成功。



- 2.3 对教学管理系统与 2011 年 6 月 2?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时间点恢复。
 - 对系统进行差异备份:

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

C:\ProgramData\MySQL\MySQL Server 5.7\Data\mylog>mysqldump -uroot -p jxgl -F>jxglf.sql Enter password: ***********							
DB.000001	2020/12/4 17:20	000001 文件	1 KB				
DB.index	2020/12/4 17:20	INDEX 文件	1 KB				
ixglf jxglf	2020/12/4 17:20	SQL Text File	5,305 KB				

修改数据库: (注意:一定要一个个执行!不能一起执行!)

update sc set grade = 93 where cno = 1 and sno = "2005001"; insert into student values("2005006","Maiqy",19,"F","Ma");



在 my.ini 中启用日志,并重启服务器:

```
# ***** Group Replication Related *****

# Specifies the base name to use for binary log files. With binary logging

# enabled, the server logs all statements that change data to the binary

# log, which is used for backup and replication.

# log-bin

log-bin="C:/ProgramData/MySQL/MySQL Server 5.7/Data/mylog/DB.log"
```

备份 mysqldump 开始以后生成的 binlog: (其实是相当于多备份一份)

mysqladmin -uroot -p flush-logs

C:\ProgramData\MySQL\MySQL Server 5.7\Data\mylog>mysqladmin -uroot -p flush-logs Enter password: ************

目录文件: (在上面的地址中)

DB.000001	2020/12/4 17:23	000001 文件	1 KB
DB.000002	2020/12/4 17:23	000002 文件	1 KB
DB.index	2020/12/4 17:23	INDEX 文件	1 KB
jxglf	2020/12/4 17:20	SQL Text File	5,305 KB

查看目录文件

mysqlbinlog -v DB.000001

执行 mysql 导入备份文件:

mysql -u root -p jxgl<jxglf.sql

C:\ProgramData\MySQL\MySQL Server 5.7\Data\mylog>mysql -u root -p jxgl<jxglf.sql Enter password: ************

数据库恢复到之前的样子:



时间点恢复:

查看出错步骤信息如下:

第一个更新 update 语句的出错信息:

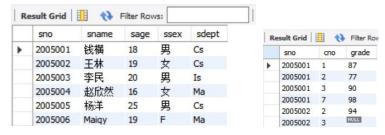
第二个插入 insert 语句的出错信息:

实现恢复:

 $mysqlbinlog \ \hbox{--no-defaults --stop-datetime} = "2020-12-04\ 17:21:14"\ DB.000001|mysql\ \hbox{--u root --h localhost --p}$

mysqlbinlog --no-defaults --start-datetime="2020-12-04 17:21:20" DB.000001|mysql -u root -h localhost -p

可以发现,跳过了更新 update,但 insert 实现了恢复,即实现了时间点的恢复:



2.4 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。

位置更新点如下:

基本步骤与上面一样,只是在位置恢复时:

mysqlbinlog --no-defaults --stop-position=293 DB.000001|mysql -u root -h localhost -p mysqlbinlog --no-defaults --start-position=441 DB.000001|mysql -u root -h localhost -p

恢复结果如下:恢复成功!

