# 实验7：数据库备份还原、日志管理与查询性能查看

**实验目的：**

1. 掌握MySQL数据库备份和恢复操作。

2. 熟悉日志操作。

3. 练习使用备份结合二进制日志恢复数据库。

**实验准备工作：**

在SQLyog中，在“root@localhost”上右击，选择执行SQL脚本，再选择cjgl.sql，创建成绩管理数据库cjgl。在SQLyog中，在“root@localhost”上右击，选择执行SQL脚本，再选择schoolinfo.sql，创建schoolinfo数据库。

**实验内容与步骤：**

**一、日志启动与查看操作**

1. 执行命令，查看系统错误日志存放位置是.\LAPTOP-96KTID8V.err

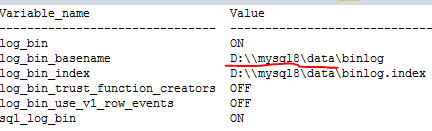
SHOW VARIABLES LIKE 'log\_error';

2. 执行命令：SHOW VARIABLES LIKE '%log\_bin%';

查看二进制日志是否开启？\_\_\_\_\_开启\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

查看二进制日志的存放位置？C:\ProgramData\MySQL\MySQL Server 8.0\Data\binlog

提示：我的电脑上是D:\mysql8\data，你的电脑上可能不同，请注意观察



3. 执行命令，查看慢日志是否开启？\_\_\_\_\_\_\_\_\_\_未开启\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

SHOW VARIABLES LIKE 'slow\_query%';

Variable\_name Value

------------------- --------------------------

slow\_query\_log OFF

slow\_query\_log\_file LAPTOP-96KTID8V-slow.log

**二、结合二进制日志还原数据库（请仔细按步骤进行操作，回答问题和记录结果）**

1.在SQLyog中查看schoolinfo数据库中有哪些表：teacherinfo

2.将schoolinfo数据库备份到 c:/bak\_schoolinfo.sql 文件中：

mysqldump -uroot -p -F schoolinfo > C:/bak\_schoolinfo.sql

注：（1）注意这一步的备份命令mysqldump是bin目录下的可执行文件；（2）由于使用了-F选项，当备份工作刚开始时系统会刷新binlog日志。

3.查看二进制日志刷新情况

SHOW MASTER LOGS;

show master status;

SHOW BINLOG EVENTS IN 'binlog.000039 '; -- 请填入最新（即最下面）的日志文件名，下面假定为XXX

4.创建一个学生信息表：

CREATE TABLE IF NOT EXISTS schoolinfo .stuinfo (

`id` int(10) unsigned NOT NULL AUTO\_INCREMENT,

`name` varchar(16) NOT NULL,

`sex` enum('m','w') NOT NULL DEFAULT 'm',

`age` tinyint(3) unsigned NOT NULL,

`classid` char(6) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

5.向学生信息表插入一些数据

insert into schoolinfo.stuinfo(`name`,`sex`,`age`,`classid`)

values('yiyi','w',20,'cls1'),

('xiaoer','m',22,'cls3'),

('zhangsan','w',21,'cls5'),

('lisi','m',20,'cls4'),

('wangwu','w',26,'cls6');

6.在SQLyog中查看此时schoolinfo数据库中有哪些表：stuinfo，teacherinfo

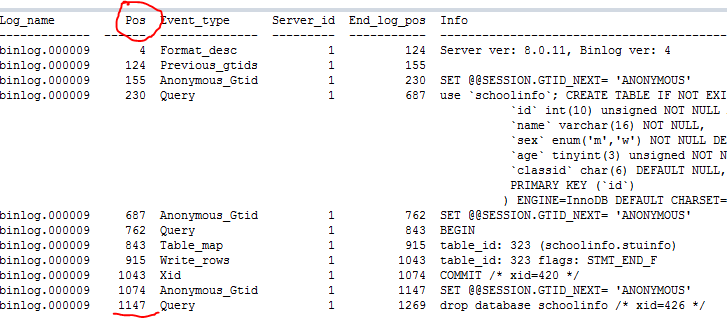
7.删除数据库

drop database schoolinfo;

8.执行： SHOW BINLOG EVENTS IN 'binlog.000039 '; 仔细查看binlog日志文件XXX的最后处，找到并记录删除数据库语句pos点（时间点）：1147（假设pos点为????）

提示：我的结果如下供参考：【1147】

SHOW BINLOG EVENTS IN 'binlog.000009'



9.执行刷新日志操作，重新开始新的binlog日志记录文件（使刚才的日志文件不会再有后续写入了）

flush logs;

show master status;

10.读取binlog日志，分析问题

show binlog events IN ' binlog.000039; -- XXX是第2步产生的日志文件

11.把备份的数据恢复：【注意这一步的mysqladmin、mysql是bin目录下的可执行文件】；

mysqladmin -uroot -p CREATE schoolinfo

mysql -u root -p schoolinfo < C:/bak\_schoolinfo.sql

（至此截至备份时刻的备份数据都恢复了）

12.在SQLyog中查看此时schoolinfo数据库中有哪些表：\_\_\_\_\_ teacherinfo

13.从binlog日志恢复数据——从指定pos结束点做恢复(部分恢复)

mysqlbinlog --stop-position=**？？？？** --database=schoolinfo **此处填上第3步的日志文件的绝对路径**| mysql -uroot -p schoolinfo

（提示：例如mysqlbinlog --stop-position=1147 --database=schoolinfo D:/MySQL8/data /binlog.000009 | mysql -uroot -p schoolinfo）

mysqlbinlog --stop-position=1147 --database=schoolinfo "C:\ProgramData\MySQL\MySQL Server 8.0\Data\binlog.000039" | mysql -uroot -p schoolinfo

14. 在SQLyog中查看此时schoolinfo数据库中有哪些表： stuinfo，teacherinfo

**三、数据导出导入操作**

1．执行如下语句，并记下查询结果：

Use cjgl;

SELECT \* FROM sc;

sno cno grade

---------- -------- --------

1114070116 11110930 65

1114070116 18130320 90

1411855228 18132220 96

1411855321 11110470 69

1412855223 18130320 60

1412855223 58130540 77

1414855302 11110140 90

1414855328 18130320 96

1414855328 58130540 85

1414855406 11110470 86

1414855406 18110140 75

1414855406 18132220 84

1418855232 18110140 87

1418855232 58130540 91

2．写出并执行SELECT…INTO OUTFILE来导出sc表中的记录，存储在c:\backup文件夹下sc\_data.txt中，其中字段分隔用，、字符串界符用”、记录行分隔用#。

SELECT \* FROM cjgl.sc

INTO OUTFILE 'c:/backup/backupdata\_sc.txt'

FIELDS TERMINATED BY ','

OPTIONALLY ENCLOSED BY '"' LINES TERMINATED BY '#';

3．先删除sc表的全部数据，然后写出并执行load data语句向sc表导入数据。

DELETE FROM cjgl.sc;

LOAD DATA INFILE 'c:/backup/backupdata\_sc.txt'

INTO TABLE cjgl.sc

FIELDS TERMINATED BY ','

OPTIONALLY ENCLOSED BY '"'

LINES TERMINATED BY '#';

**四、记录下列查询语句的查找行数**

USE cjgl;

EXPLAIN SELECT \* FROM cjgl.student WHERE sname='孙一凯'; -- rows= \_\_\_\_16\_\_\_\_\_\_\_\_

CREATE INDEX idx\_sname ON cjgl.student(sname); -- rows= \_\_\_\_\_无\_\_\_\_\_\_\_

EXPLAIN SELECT \* FROM cjgl.student WHERE sname='孙一凯'; -- rows= \_\_\_\_\_\_\_1\_\_\_\_\_

EXPLAIN SELECT \* FROM cjgl.student WHERE sname='孙'; -- rows= \_\_\_\_\_1\_\_\_\_\_\_\_

EXPLAIN SELECT \* FROM cjgl.student WHERE sname LIKE '%孙'; -- rows= \_\_\_\_\_\_\_\_\_\_16\_\_

EXPLAIN SELECT \* FROM cjgl.student WHERE sname LIKE '孙%'; -- rows= \_\_\_\_1\_\_\_\_\_\_\_\_

EXPLAIN SELECT \* FROM cjgl.student WHERE sname='孙一凯'OR zno='1309070338'; -- rows= \_\_2\_\_\_\_\_

EXPLAIN SELECT \* FROM cjgl.student WHERE sname='孙一凯'OR ssex='男'; -- rows= \_\_\_\_\_16\_\_\_\_\_\_\_

EXPLAIN SELECT sname FROM student WHERE sno IN (SELECT sno FROM sc WHERE grade>90); -- rows= \_\_null,1,14\_

EXPLAIN SELECT sname FROM student INNER JOIN sc USING(sno) WHERE grade>90; -- rows= \_14,1\_\_\_\_\_

**五、事务应用：创建存储过程，模拟银行转账事务**

1.创建以你自己姓名全拼命名的数据库，然后选用该数据库成为当前数据库

CREATE DATABASE chenlong;

2.执行下面的语句，创建account表，并添加数据

CREATE TABLE account(aID CHAR(10) PRIMARY KEY,aname CHAR(10),balance INT );

INSERT INTO account VALUES('001','张三',30000);

INSERT INTO account VALUES('002','李四',50000);

3.创建存储过程transferMoney，有三个输入参数：A\_account、B\_account、money，存储过程完成的功能是由A账户转给B账户money元。（**要求运用事务**）

DELIMITER $$

CREATE PROCEDURE transferMoney

(IN A\_account CHAR(10),IN B\_account CHAR(10),IN money INT(10) )

BEGIN

START TRANSACTION;

UPDATE account SET balance=balance-money WHERE aID=A\_account;

UPDATE account SET balance=balance+money WHERE aID=B\_account;

COMMIT;

END $$

DELIMITER ;

4.调用执行存储过程transferMoney，由'002'账户转给'001'账户500元。

CALL transferMoney('002','001',500)

**实验总结：实验中遇到的主要问题及解决方法：**

1. **在进行日志恢复时，首先在命令提示符中要注意大小写，在输入导入binlog文件路径时要注意在路径上加 ” ”,否则识别不了。**
2. **在SELECT…INTO OUTFILE导出表中的记录时，要提前修改my.ini里面的secure-file-priv="C:/ProgramData/MySQL/MySQL Server 8.0/Uploads"为 secure-file-priv=**
3. **mysqldump -uroot -p -F schoolinfo > C:/bak\_schoolinfo.sql要在命令提示符里面mysql/bin下进行，因为只有bin里面才有mysqldump命令**