MySQL 数据备份与恢复实验

一【实验目标】

- 了解什么是数据备份与恢复
- 掌握使用 MySQL 进行数据库的备份与恢复

二【实验环境】

- Windows 10
- MySQL

三【实验原理】

数据备份就是保存数据的副本。数据备份的目的是预防事故（如自然灾害、病毒破坏和人为损坏等）造成的数据损失。

数据恢复就是将数据恢复到事故之前的状态。数据恢复总是与备份相对应。备份是恢复的前提，恢复是备份的目的，无法恢复的备份是没有意义的。

一、使用 SQL 语句备份和恢复

1. 用户可以使用 SELECT INTO…OUTFILE 语句把表数据导出到一个文本文件中，并用 LOAD DATA …INFILE 语句恢复数据。但是这种方法只能导出或导入数据的内容，不包括表的结构，如果表的结构文件损坏，则必须先恢复原来的表的结构。

SELECT INTO…OUTFILE 格式：

```
SELECT * INTO OUTFILE 'file_name' export_options
| DUMPFILE 'file_name'
```

其中，export_options 为：

```
[FIELDS
    [TERMINATED BY 'string']
    [[OPTIONALLY] ENCLOSED BY 'char']
    [ESCAPED BY 'char' ]
]
    [LINES TERMINATED BY 'string' ]
```

说明：这个语句的作用是将表中被 SELECT 语句选中的行写入到一个文件中，

file_name 是文件的名称。文件默认在服务器主机上创建，并且文件名不能是已经存在的（这可能将原文件覆盖）。如果要将该文件写入到一个特定的位置，则要在文件名前加上具体的路径。在文件中，数据行以一定的形式存放，空值用“\N”表示。

使用 OUTFILE 时，可以在 export_options 中加入以下两个自选的子句，它们的作用是决定数据行在文件中存放的格式：

（1） FIELDS 子句：在 FIELDS 子句中有三个亚子句：TERMINATED BY、[OPTIONALLY] ENCLOSED BY 和 ESCAPED BY。如果指定了 FIELDS 子句，则这三个亚子句中至少要指定一个。

- TERMINATED BY 用来指定字段值之间的符号，例如，TERMINATED BY ’,’ 指定了逗号作为两个字段值之间的标志。
- ENCLOSED BY 子句用来指定包裹文件中字符值的符号，例如，ENCLOSED BY ’ ″ ’表示文件中字符值放在双引号之间，若加上关键字 OPTIONALLY 表示所有的值都放在双引号之间。
- ESCAPED BY 子句用来指定转义字符，例如，ESCAPED BY ’*’将*指定为转义字符，取代“\”，如空格将表示为“*N”。

（2） LINES 子句：在 LINES 子句中使用 TERMINATED BY 指定一行结束的标志，如 LINES TERMINATED BY ’?’表示一行以“?”作为结束标志。如果 FIELDS 和 LINES 子句都不指定，则默认声明以下子句：

```
FIELDS TERMINATED BY ’\t’ ENCLOSED BY ’’ ESCAPED BY ’\\’
LINES TERMINATED BY ’\n’
```

如果使用 DUMPFILE 而不是使用 OUTFILE，导出的文件里所有的行将彼此紧挨着放置，值和行之间没有任何标记，成了一个长长的值。

2. LOAD DATA …INFILE 语句是 SELECT INTO…OUTFILE 语句的补语，该语句可以将一个文件中的数据导入到数据库中。

LOAD DATA …INFILE 格式：

```
LOAD DATA [LOW_PRIORITY | CONCURRENT] [LOCAL] INFILE
’file_name.txt’
[REPLACE | IGNORE]
```

```
    INTO TABLE tbl_name
[FIELDS
   [TERMINATED BY 'string']
   [[OPTIONALLY] ENCLOSED BY 'char']
   [ESCAPED BY 'char' ]
]
[LINES
   [STARTING BY 'string']
   [TERMINATED BY 'string']
]
[IGNORE number LINES]
[(col_name_or_user_var,...)]
[SET col_name = expr,...)]
```

说明： LOW_PRIORITY | CONCURRENT：若指定 LOW_PRIORITY，则延迟语句的执行。若指定 CONCURRENT，则当 LOAD DATA 正在执行的时候，其他线程可以同时使用该表的数据。

二、使用 mysqlimport 恢复数据

利用 mysqlimport 工具恢复数据，它完全是与 LOAD DATA 语句对应的，由发送一个 LOAD DATA INFILE 命令到服务器来运作。

mysqlimport 命令格式为：

mysqlimport [options] db_name filename ...

说明：options 是 mysqlimport 命令的选项，使用 mysqlimport -help 即可查看这些选项的内容和作用。常用的选项为：

-d，--delete：在导入文本文件前清空表格。

--lock-tables：在处理任何文本文件前锁定所有的表。这保证所有的表在服务器上同步。而对于 InnoDB 类型的表则不必进行锁定。

--low-priority，--local，--replace，--ignore：分别对应 LOAD DATA INFILE 语句的 LOW_PRIORITY，LOCAL，REPLACE，IGNORE 关键字。

对于在命令行上命名的每个文本文件，mysqlimport 剥去文件名的扩展名，

并使用它决定向哪个表导入文件的内容。例如，“patient.txt”、“patient.sql”和“patient”都会被导入名为patient的表中。所以备份的文件名应根据需要恢复表命名。

三、使用mysqldump备份数据

利用mysqldump工具备份数据，但是它比SQL语句多做的工作是可以在导出的文件中包括SQL语句，因此可以备份数据库表的结构，而且可以备份一个数据库，甚至整个数据库系统。

mysqldump [OPTIONS] database [tables]

mysqldump [OPTIONS] --databases [OPTIONS] DB1 [DB2 DB3...]

mysqldump [OPTIONS] --all-databases [OPTIONS]

如果不给定任何表，整个数据库将被倾倒。通过执行mysqldump --help，能得到mysqldump的版本支持的选项表。

若备份数据库 db_name，语法格式为 shell> mydqldump db_name。由于mysqldump缺省时把输出定位到标准输出，所以可根据需要重定向标准输出，语法格式为shell> mydqldump db_name>db_name.bak。

同其他客户机一样，必须提供一个MySQL数据库帐号用来导出数据库，如果不是使用匿名用户的话，可能需要手工提供参数或者使用选项文件：shell>mysql -u root －pmypass db_name>db_name.sql。

四、用直接拷贝的方法备份恢复

由于MySQL的数据库和表是直接通过目录和表文件实现的，因此直接复制文件来备份数据库数据，对MySQL来说特别方便。而且自MySQL 3.23起MyISAM表成为缺省的表的类型，这种表为在不同的硬件体系中共享数据提供了保证。

使用直接拷贝的方法备份时，尤其要注意表如果没有被使用，应该首先对表进行读锁定。

备份一个表，需要三个文件：描述文件、数据文件和索引文件。

四【实验步骤】

1、输入密码【Admin123456】，登录系统。

2、打开 MYSQL 命令行程序，密码 111 先建立数据库：`create database mytest;`，切换到数据库后：`use mytest;`，输入下面的命令创建一个 students 表。

```
create table students (
id int auto_increment not null primary key,
name char(20),
score int);
```

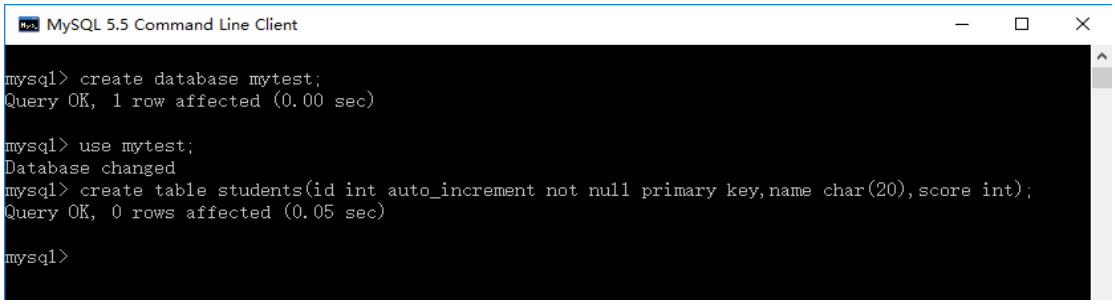

图 1

3、插入三条记录。

```
insert into students values('1','aaaa','85');
insert into students values('2','bbbb','88');
insert into students values('3','cccc','90');
```

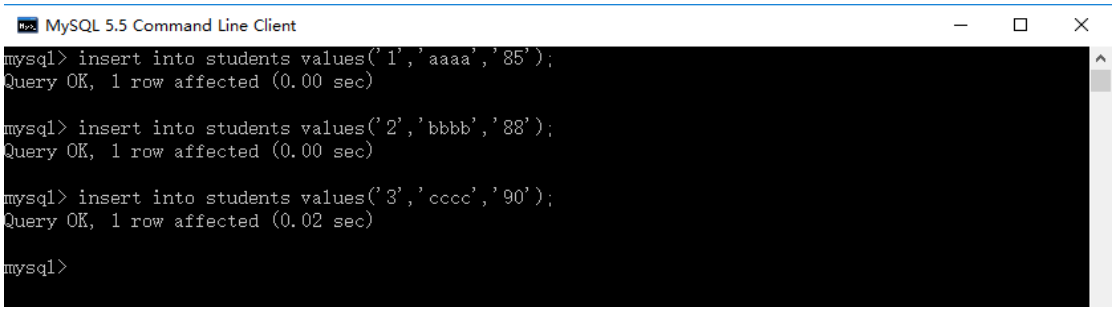

图 2

4、将建立好的 students 表备份到指定目录文本文件中。

```
select * from students into outfile 'C:/ProgramData/MySQL/MySQL
Server 5.5/Uploads/students_backup.txt'
FIELDS TERMINATED BY ','
```

OPTIONALLY ENCLOSED BY '"'

LINES TERMINATED BY '?';

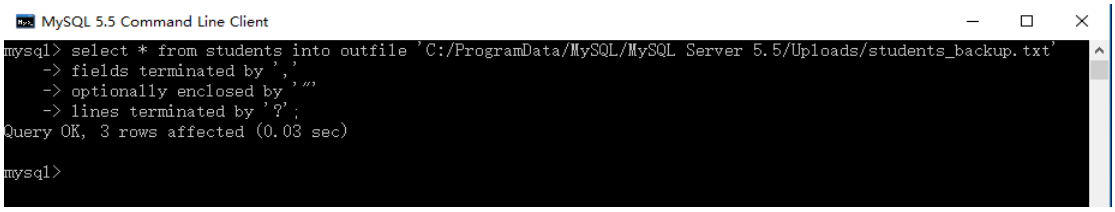

图 3

5、查看文本文件备份内容。

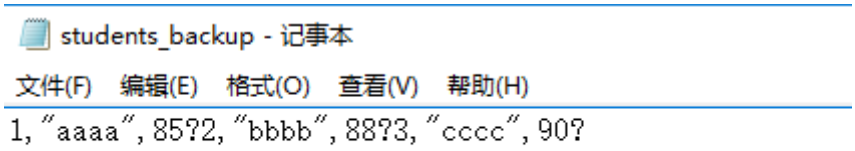

图 4

6、删除两条记录,查看此时的数据表内容。

delete from students where id='2';

delete from students where id='3';

select * from students;

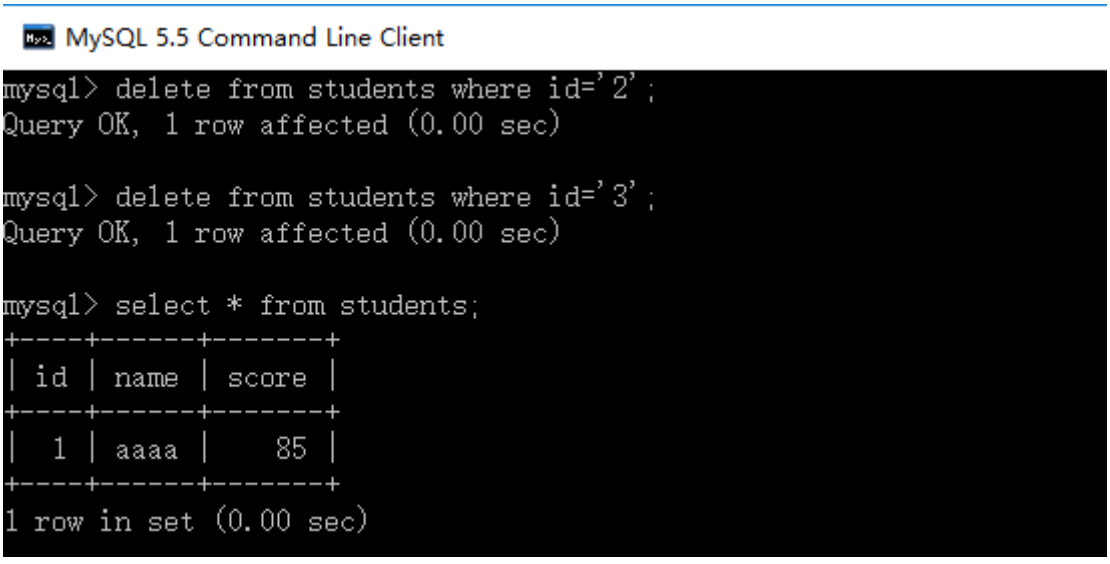

图 5

7、从文件中恢复数据表内容，查看此时数据表内容，发现数据成功恢复。

load data infile 'C:/ProgramData/MySQL/MySQL Server 5.5/Uploads/students_backup.txt' replace into table students

FIELDS TERMINATED BY ’,’

OPTIONALLY ENCLOSED BY ’”’

LINES TERMINATED BY ’?’;

select * from students;

```
mysql> load data infile 'C:/ProgramData/MySQL/MySQL Server 5.5/Uploads/students_backup.txt' replace into table students
    -> fields terminated by ','
    -> optionally enclosed by '"'
    -> lines terminated by '?';
Query OK, 3 rows affected (0.00 sec)
Records: 3  Deleted: 0  Skipped: 0  Warnings: 0

mysql> select * from students;
+----+------+-------+
| id | name | score |
+----+------+-------+
|  1 | aaaa |    85 |
|  2 | bbbb |    88 |
|  3 | cccc |    90 |
+----+------+-------+
3 rows in set (0.00 sec)
```

图 6

8、使用 MySQL 中自带的 mysqldump 程序备份数据库，在命令行中切换到 mysqldump 程序所在目录，输入如下命令将 mytest 数据库备份到 mytest.sql 文件中。其中-u 参数为用户，-p 参数为密码。

mysqldump -u root -p111 mytest >"文件路径"

```
管理员: 命令提示符

C:\Tools\mysql\bin>mysqldump -u root -p111 mytest >"C:\ProgramData\MySQL\MySQL Server 5.5\Uploads\mytest.sql"

C:\Tools\mysql\bin>
```

图 7

9、与之前一样删除一个记录。

```
MySQL 5.5 Command Line Client
mysql> delete from students where id='3';
Query OK, 1 row affected (0.00 sec)

mysql> select * from students;
+----+------+-------+
| id | name | score |
+----+------+-------+
|  1 | aaaa |    85 |
|  2 | bbbb |    88 |
+----+------+-------+
2 rows in set (0.00 sec)
```

图 8

10、从备份中恢复数据库，这里同样需要输入用户名和密码。

mysql -u root -p111 mytest <"文件路径"

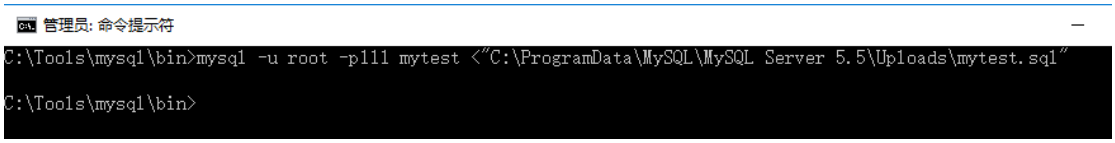

图 9

11、查看恢复成功后的数据库信息，与删除记录前一样，备份恢复成功。

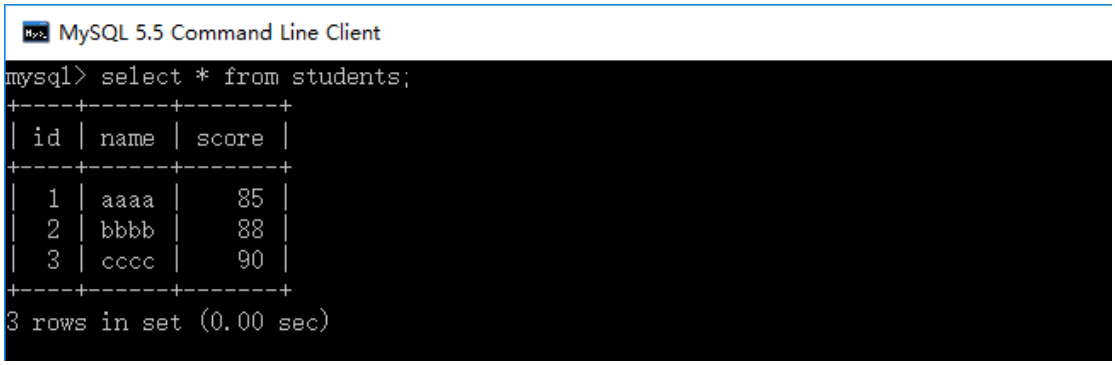

图 10

五【实验思考】

- 你还知道哪些数据库备份恢复方法？