**电 子 科 技 大 学**

**实 验 报 告**

**学生姓名：赵则橦 学 号：2017221305013**

**指导教师: 钟婷**

**实验地点：信软400 实验时间：2019/11/17**

**一、实验室名称：** 网络安全专业课程实验室

**二、实验项目名称：** MySQL安全配置实验

**三、实验学时：** 2

**四、实验原理：**

数据库管理系统内置了安全配置，安全审计，数据库备份与恢复的相关功能。

**五、实验目的：**

1. 掌握MySQL安全配置的关键技术。
2. 掌握MySQL安全审计的关键技术。
3. 掌握MySQL数据库备份与恢复的关键技术。

**六、实验内容：**

1. MySQL安全配置
2. MySQL安全审计
3. MySQL数据库备份与恢复

**七、实验器材（设备、元器件）：**

PC微机一台、SimpleISES信息安全实验教学系统

**八、实验步骤：**

1. MySQL安全配置

打开Windows实验台，运行Windows 2003系统；运行MySQL的命令窗口。

**注：以下步骤中所涉及的路径及文件名，应根据实验环境及需要进行相应调整。**

1. **查看User表**

运行如下图所示的命令，查看User表。

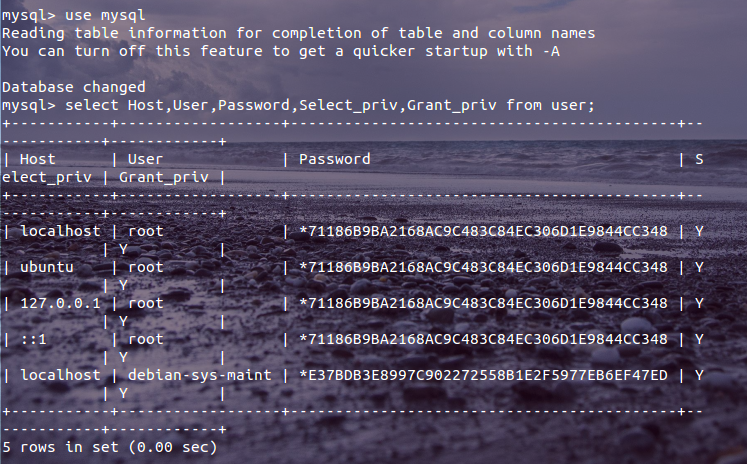


图2.2.2‑1

1. **修改用户root 的密码**

运行如下图所示的命令，将MySQL数据库root用户的口令改成wsxcokmn1234。其中最后一句命令flush privileges的意思是强制刷新内存授权表，否则用的还是缓冲中的口令，这时非法用户还可以用root用户及空口令登录，直到重启MySQL服务器。

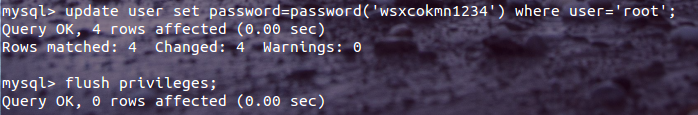


图2.2.2‑2

1. **删除空用户**

运行如下图所示的命令，删除空用户。Host字段为localhost的匿名用户拥有所有的权限，就是说本地用户用空的用户名和空的口令登录MySQL数据库服务器可以得到最高的权限，所以匿名用户必须删除。

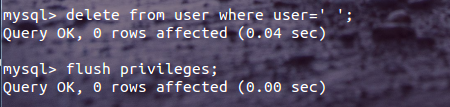


图2.2.2‑3

1. **MySQL授权表的结构与内容**
2. Desc user

User表是授权表中最重要的一个，运行如下图所示的命令，列出可以连接服务器的用户及其加密口令，并且它指定它们有哪种全局(超级用户)权限。在User表启用的任何权限均是全局权限，并适用于所有数据库。所以我们不能给任何用户访问mysql.user表的权限。

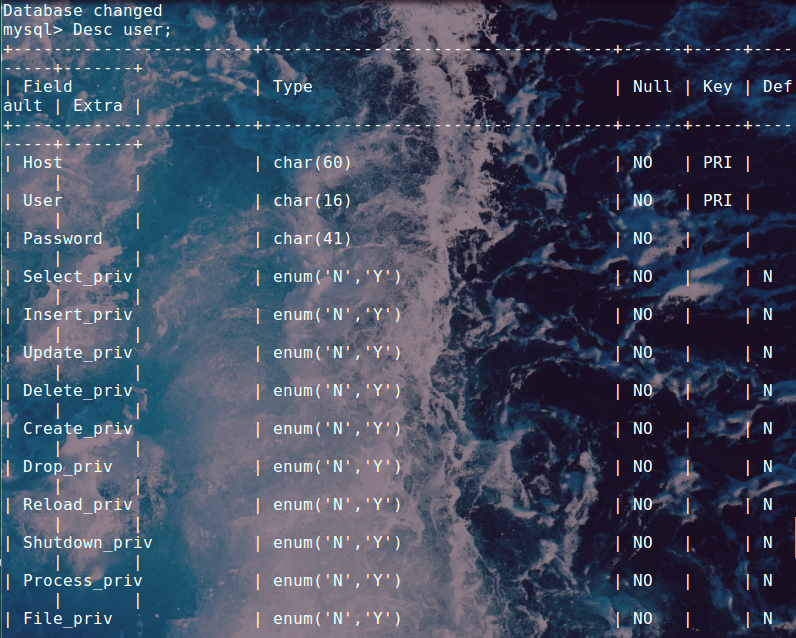


图2.2.2‑4

具体的权限说明如下图所示。

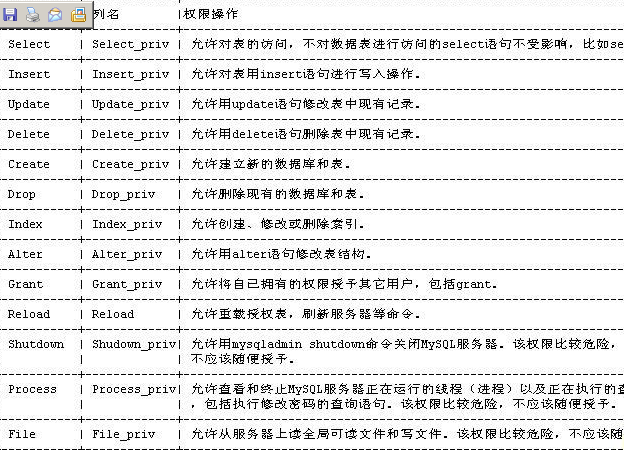


图2.2.2‑5

1. Desc db

运行如下图所示的命令，查看Db表，列出所有数据库，而用户有权限访问它们。在这里指定的权限适用于一个数据库中的所有表。



图2.2.2‑6

1. Desc host

Host表如下图所示，与db表结合使用在一个较好层次上控制特定主机对数据库的访问权限，这可能比单独使用db好些。这个表不受GRANT和REVOKE语句的影响。

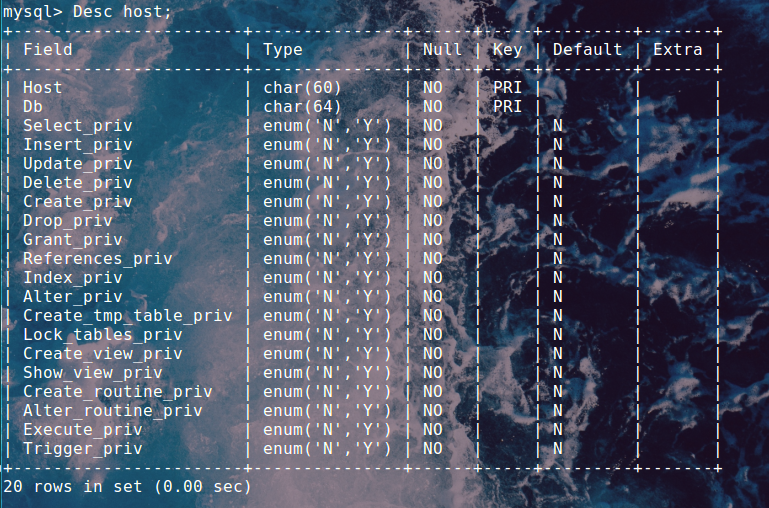


图2.2.2‑7

1. **增加用户，并授权**

如下图所示的命令用于增加一个本地具有所有权限的test用户(超级用户)，密码是test。ON子句中的\*.\*意味着“所有数据库、所有表”。with grant option表示它具有grant权限。用grant语句创建权限是不需要再手工刷新授权表的，因为它已经自动刷新了。

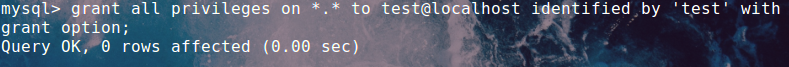


图2.2.2‑8

查看运行结果，如下图所示。

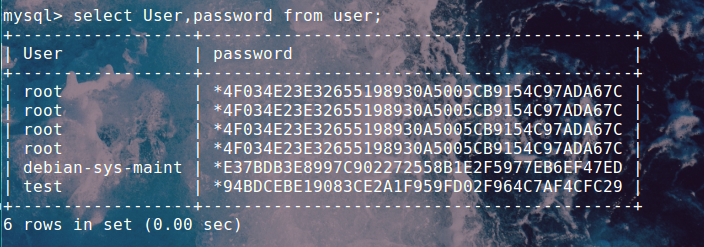


图2.2.2‑9

1. **删除用户的权限，并删除用户**

删除用户权限，如下图所示:



图2.2.2‑10

删除用户test，如下图所示。

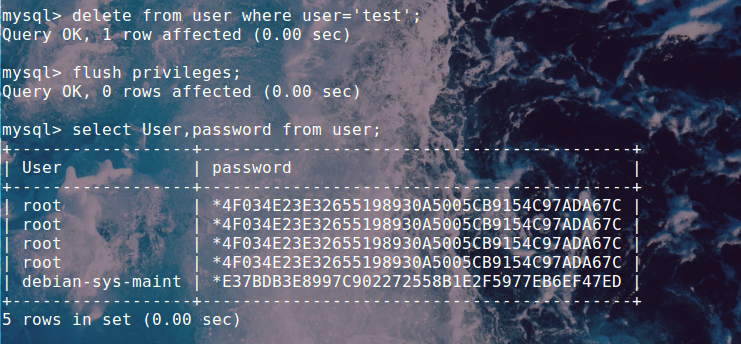


图2.2.2‑11

1. MySQL安全审计

打开Windows实验台，运行Windows 2003系统。

**注：以下步骤中所涉及的路径及文件名，应根据实验环境及需要进行相应调整。**

1. **日志文件的配置及查看**
2. 启动mysql 的日志

启动mysql的日志，对其安装目录下的my.ini文件进行配置：

1. 错误日志

My.ini文件中配置信息如下：

#Enter a name for the error log file. Otherwise a default name will be used.

#log-error= D:/MySQL/MySQLServer 5.0/data/log/error/mysql\_log\_err.txt

1. 查询日志

My.ini文件中配置信息如下：

#Enter a name for the query log file. Otherwise a default name will be used.

#log= D:/MySQL/MySQLServer 5.0/data/log/query/mysql\_log.txt

1. 更新日志

My.ini文件中配置信息如下：

#Enter a name for the update log file. Otherwise a default name will be used.

#log-update=D:/MySQL/MySQLServer 5.0/data/log/update/mysql\_log\_update.txt

1. 二进制日志

My.ini文件中配置信息如下：

#Enter a name for the binary log. Otherwise a default name will be used.

#log-bin= D:/MySQL/MySQLServer 5.0/data/log/binary/mysql\_log\_bin

1. 慢日志

My.ini文件中配置信息如下：

#Enter a name for the slow query log file. Otherwise a default name will be used.

#long\_query\_time =1

#log-slow-queries=D:/MySQL/log/slow/mysql\_log\_slow.txt

1. 查看Mysql日志的关闭与开启

如下图所示，凡Value值为OFF的表示未开启服务，若要开启只需要将上述的my.ini配置信息写入(my.ini为mysql安装目录下的配置文件)，然后去掉前面的“#”号，再重启mysql服务。现在会看到指定的日志文件已创建。相反地，若要停止mysql日志服务，只需要将my.ini中对应的配置信息去掉即可。

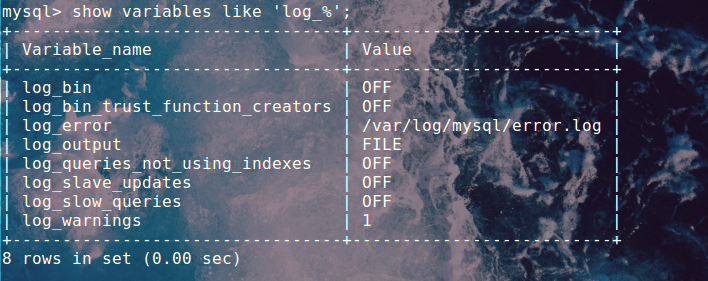


图2.2.2‑26

1. **管理和使用mysql的二进制日志**

从实验原理中可以看到my.ini配置信息的log-bin没有指定文件扩展名，这是因为即使你指定上扩展名它也不使用。当mysql创建二进制日志文件时，首先创建一个以“mysql\_log\_bin”为名称，以“.index”为后缀的文件；再创建一个以“mysql\_log\_bin”为名称，以“.000001”为后缀的文件。当mysql服务重新启动一次以“.000001”为后缀的文件会增加一个，并且后缀名加1递增；如果日志长度超过了max\_binlog\_size的上限(默认是1G)也会创建一个新的日志文件；使用flushlogs(mysql命令符)或者执行mysqladmin –u –p flush-logs(Windows命令提示符)也会创建一个新的日志文件。日志中写入的是二进制数据，用记事本打开文件是看不到正常数据的，使用如下命令对二进制日志进行管理操作：

1. 使用BIN目录下的mysqlbinlog命令，查看日志

输入命令：mysqlbinlog D:\mysql\log\binary\mysql\_log\_bin.000001；运行结果如下图所示。

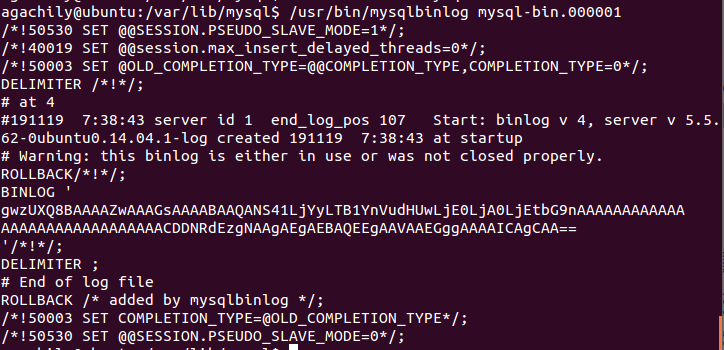
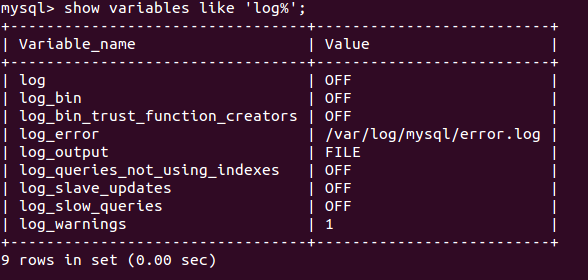


图2.2.2‑27

1. 使用SQL语句查看日志

输入命令：show master logs; ，运行结果如下图所示。

先查看是否启用了日志：



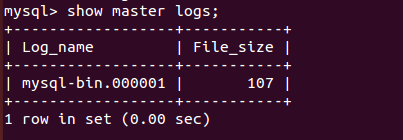


图2.2.2‑28

输入命令：show master status; ，运行结果如下图所示。

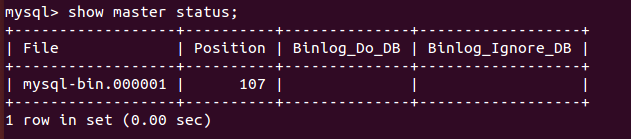


图2.2.2‑29

1. 把二进制的日志文件转换成.sql的文件

输入命令：mysqlbinlog D:\mysql\log\binary\mysql\_log\_bin.000001 >aabb.sql，如下图所示；运行结果如下：

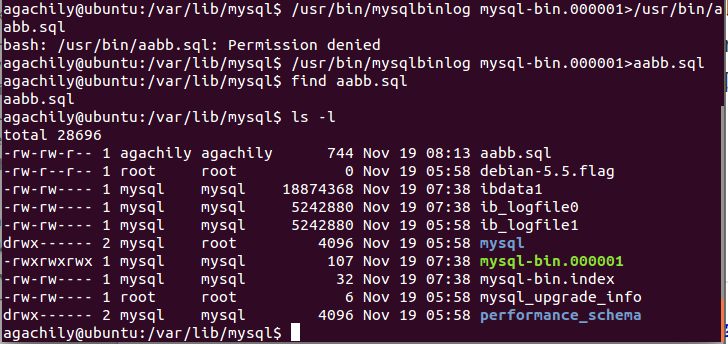


图2.2.2‑31

1. 使用日志文件恢复数据

MySQL的二进制日志文件是MySQL安全系统里的非常重要的文件，虽然打开日志，所有非SELECT的操作都会记录下来，但这在某些场合显得很冗余，比如还原数据库，或者导入数据等，日志文件会一条不漏的记录下所有的命令，这会使日志文件迅速变大。但正是因为这样，让数据库的安全性变得更加可靠。下面将通过日志文件恢复误操作，找回丢失的数据。

1. 打开数据库，查看表，创建新表student，如下图所示。



图2.2.2‑32

1. 向student表中插入数据，并查询student表，如下图所示。

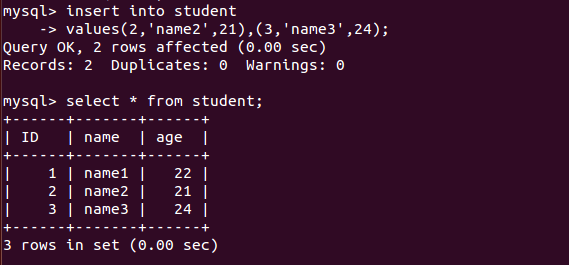


图2.2.2‑33

1. 删除student表中数据，删除表，如下图所示。

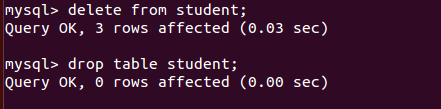


图2.2.2‑34

1. 查看二进制日志，如下图所示。

先更改权限，再进行查看：

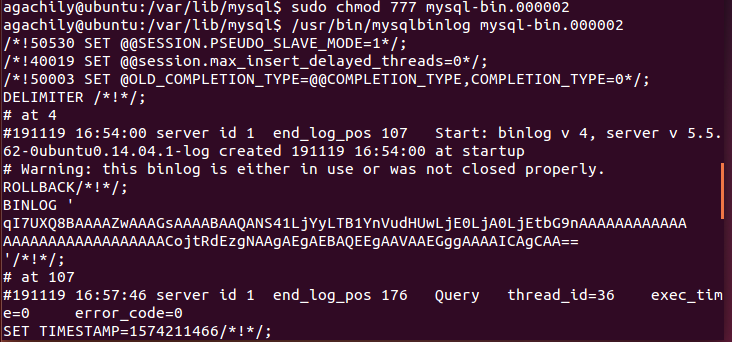


图2.2.2‑35

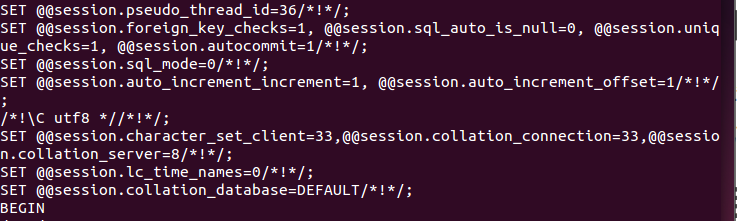
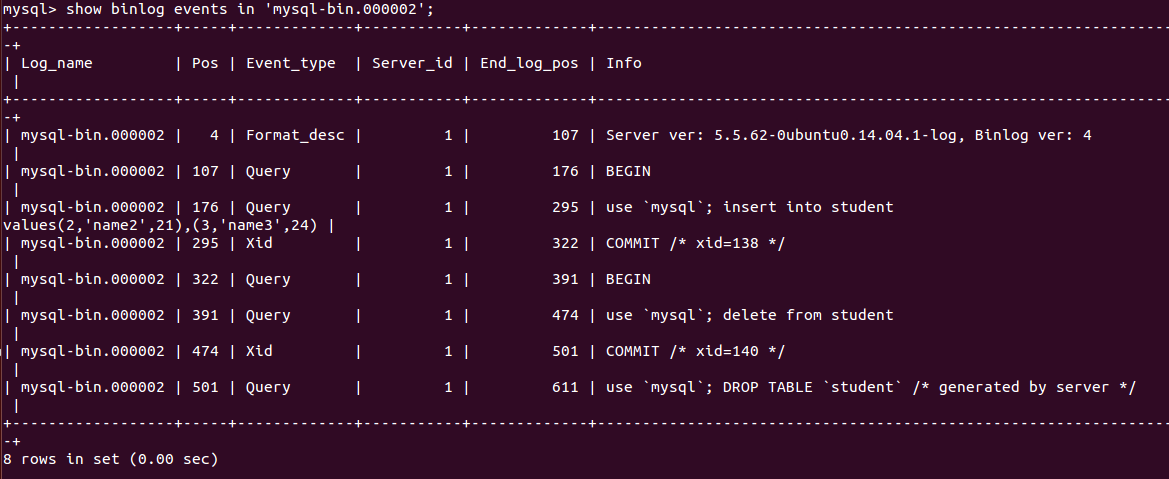


图2.2.2‑36

1. 首先查看000002的binlog日志，可以看到删除操作开始于322，因此恢复到322之前。



截取日志的第4行至第322行，存入huifu.sql文件，如下图所示。



图2.2.2‑37

1. 把提取出来的日志文件段导入到数据库中，如下图所示。

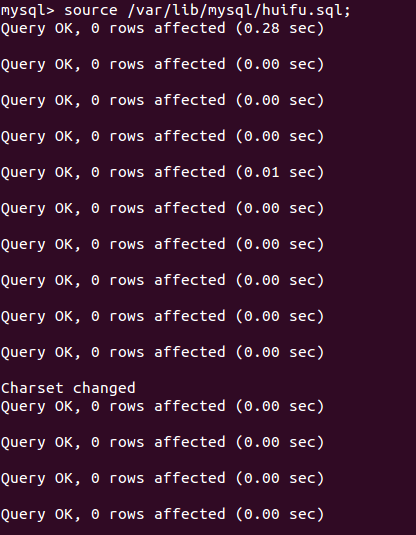


图2.2.2‑38

1. 查看恢复的结果，如下图所示。

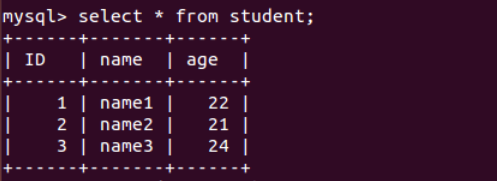


图2.2.2‑39

1. MySQL数据库备份与恢复

打开Windows实验台，运行Windows 2003系统；打开cmd.exe命令窗口。

**注：以下步骤中所涉及的路径及文件名，应根据实验环境及需要进行相应调整。**

1. **打开连接，查看版本**

打开“开始|运行”，输入“cmd”回车，进入命令窗口，通过“cd”命令把路径转换为 MySQL的安装路径下；

输入“mysql –u root –p123456”回车打开连接，如下图所示。

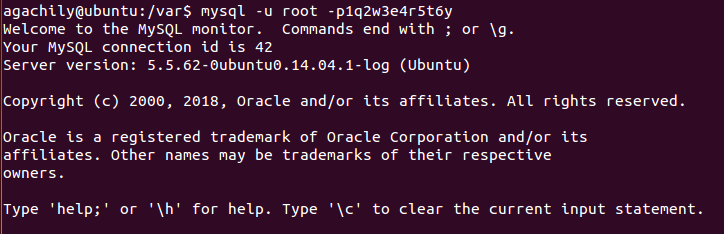


图2.2.2‑12打开连接

在提示符下输入“select version()；”查看版本号，如下图所示。

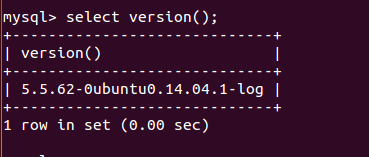


图2.2.2‑13查看版本号

1. **查询数据库列表、打开数据库、显示其中的数据表**

输入show database，查询数据库列表，如下图所示。

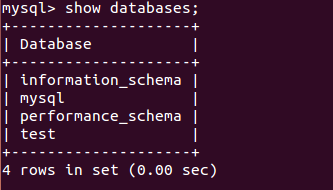


图2.2.2‑14

输入如下图所示的命令，打开数据库test并显示其中的数据表信息。

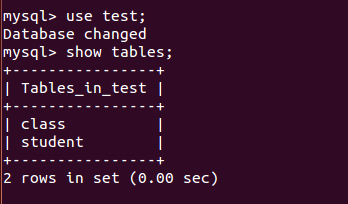


图2.2.2‑15

1. **显示表结构、查询表中数据、插入数据**

输入如下图所示的命令，显示student表的表结构。

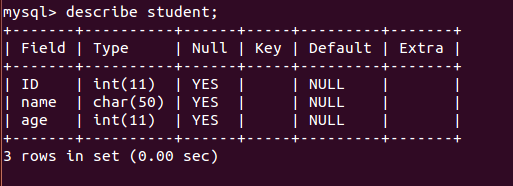


图2.2.2‑16显示表结构

输入如下图所示的SQL语句，查询student表中的数据。

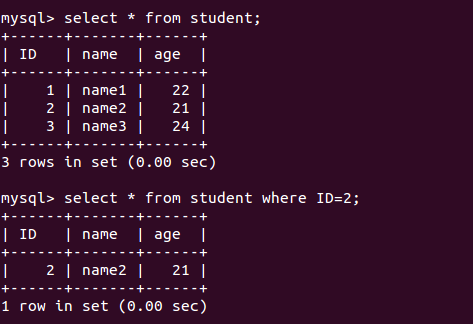


图2.2.2‑17查询记录

输入如下图所示的SQL语句，向student表中插入数据并查看运行结果。

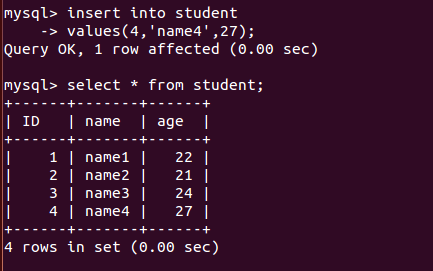


图2.2.2‑18插入数据

1. **更新记录、删除记录**

输入如下图所示的SQL语句，更新student表中的某个记录并查看运行结果。

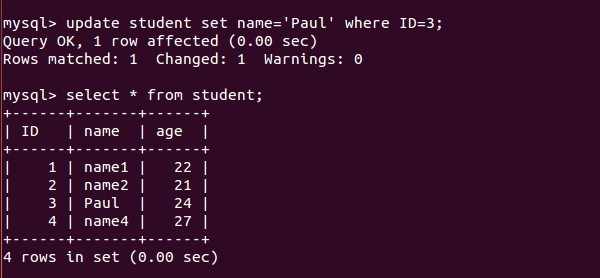


图2.2.2‑19更新记录

输入如下图所示的SQL语句，删除student表中的某个记录并查看运行结果。

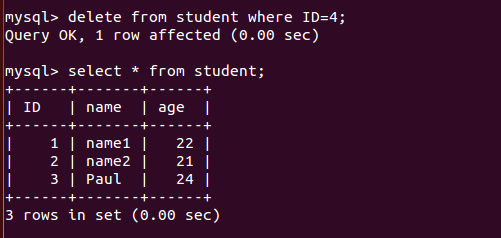


图2.2.2‑20删除记录

1. **备份数据表，还原数据**

输入如下图所示的SQL语句，备份数据表。



图2.2.2‑21备份数据表

输入如下图所示的SQL语句，删除记录后进行恢复。

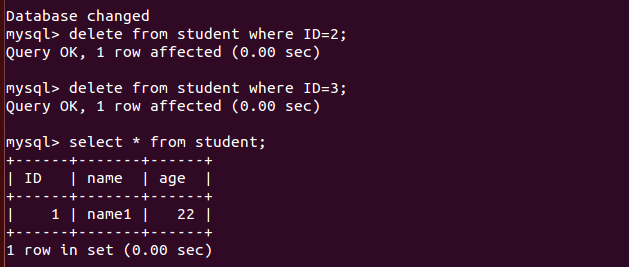




图2.2.2‑22删除记录、恢复记录

如下图所示，查看回复后的记录。

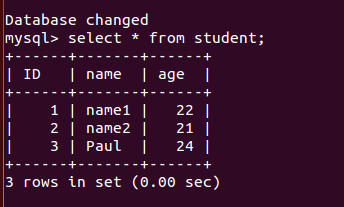


图2.2.2‑23查看记录

1. **备份和还原数据库**

输入如下图所示的SQL语句，备份数据库test。



图2.2.2‑24备份test数据库

输入如下图所示的SQL语句，还原数据库test。



图2.2.2‑25还原test数据库

**九、实验数据及结果分析：**

可以成功的实现MySQL数据库的安全配置，实现安全审计功能以及数据库的备份与恢复。

**十、实验结论：**

MySQL数据库内置了强大的安全配置，安全审计与数据库备份与恢复功能。

**十一、总结及心得体会：**

通过本实验对MySQL数据库的安全配置有了进一步的了解。

**十二、对本实验过程及方法、手段的改进建议：**

无

**报告评分： XX**

**指导教师签字：XXXX**