**《数据库原理》实验报告**

**实验名称 数据库完整性与安全性实验**

**班 级 2014211302**

**学 号 2014211182**

**姓 名 曹桢**

# 实验5 数据库完整性与安全性实验

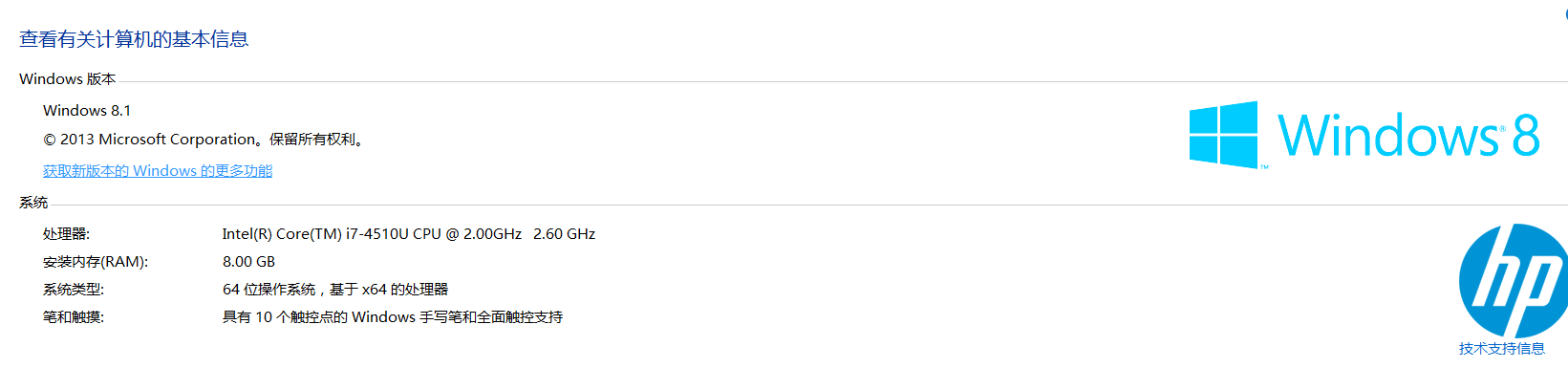
一、实验目的

1. 通过对完整性规则的定义实现，熟悉了解mysql中完整性保证的规则和实现方法，加深对数据完整性的理解。

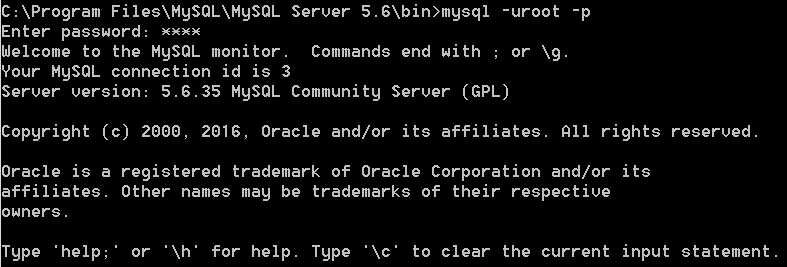
2. 通过对安全性相关内容的定义，熟悉了解mysql中安全性的内容和实现方法，加深对数据库安全性的理解

**二、实验环境**

* 操作系统环境：windows8.1 64位



* 数据库版本：mysql 5.6



**三、实验内容**

**完整性实验：**

1. 分别定义学生数据库中各基表的主键、外键，实现实体完整性约束和参照完整性约束；
2. 分别向学生表、课程表插入具有相同学号和相同课程编号的学生数据和课程数据，验证其实体完整性约束；
3. 向学生选课表中插入一条数据，课程编号是课程表中没有的，验证参照完整性约束；
4. 删除学生表中的所有数据，验证参照完整性约束；
5. 定义存储过程，完成查询某个学生的选课情况，并执行。
6. 定义触发器，当向学生表插入新的一条记录时，将所有学生出生日期加1；并对其进行测试。
7. 用sql完成以上操作。

**安全性实验内容**

1. 定义一新的登陆帐号、数据库用户，并授予其访问学生数据库的读权限；
2. 分别用sa用户和新定义的用户访问学生数据库，并对其中的学生表数据进行修改；
3. 再次用此用户访问学生数据库，并对其中的学生表数据进行修改。
4. 用SQL语句分别完成以上内容。

**四、实验步骤**

**1、完整性实验：**

（1）分别定义学生数据库中各基表的主键、外键，实现实体完整性约束和参照完整性约束；

**主键**:建表时已创建, 实现实体完整性约束。

CREATE TABLE `student` (

`sno` varchar(6) NOT NULL,

`sname` varchar(6) DEFAULT NULL,

`sex` varchar(2) DEFAULT NULL,

`bdate` datetime DEFAULT NULL,

`dept` varchar(8) DEFAULT NULL,

`classno` varchar(3) DEFAULT NULL,

PRIMARY KEY (`sno`)

);

CREATE TABLE `course` (

`cno` varchar(3) NOT NULL,

`cname` varchar(12) DEFAULT NULL,

`lhour` int(11) DEFAULT NULL,

`credit` int(11) DEFAULT NULL,

`semester` varchar(2) DEFAULT NULL,

PRIMARY KEY (`cno`)

);

CREATE TABLE `sc` (

`sno` varchar(6) NOT NULL,

`cno` varchar(3) NOT NULL,

`grade` int(11) DEFAULT NULL,

PRIMARY KEY (`sno`,`cno`)

);

**外键**: 实现参照完整性约束。

alter table sc add constraint sno

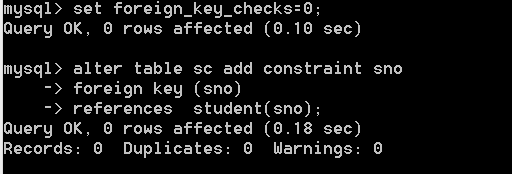
foreign key (sno)

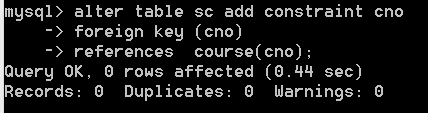
references student(sno);

alter table sc add constraint cno

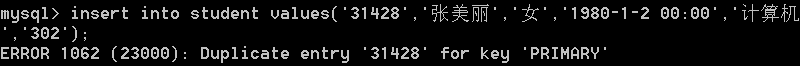
foreign key (cno)

references course(cno);





（2）分别向学生表、课程表插入具有相同学号和相同课程编号的学生数据和课程数据，验证其实体完整性约束；





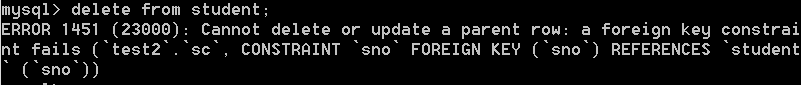
实体完整性约束验证成功:由于已经有了这些主键，所以无法插入。

（3）向学生选课表中插入一条数据，课程编号是课程表中没有的，验证参照完整性约束；



由于没有’C00’这个课程号，不能插入，参照完整性验证成功。

（4）删除学生表中的所有数据，验证参照完整性约束；



由于参照完整性约束，删除失败。

1. 定义存储过程，完成查询某个学生的选课情况，并执行。

create procedure search

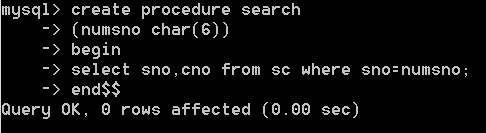
(numsno char(6))

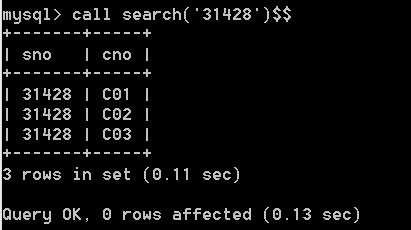
begin

select sno,cno from sc where sno=numsno;

end$$









1. 定义触发器，当向学生表插入新的一条记录时，将所有学生出生日期加1；并对其进行测试。



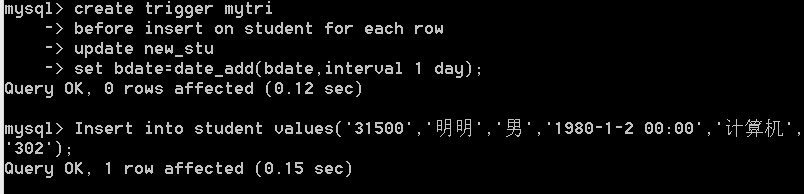
create trigger mytri

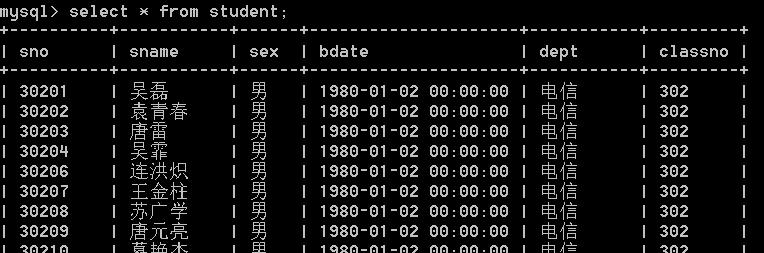
before insert on student for each row

update new\_stu

set bdate=date\_add(bdate,interval 1 day);

Insert into student values('31500','明明','男','1980-1-2 00:00','计算机','302');









查看student和new\_stu的表项，发现student中成功插入了这一元组，而new\_stu成功被触发器触发，出生日期从1980-01-02变成1980-01-03。

**2、安全性实验**

（1）定义一新的登陆帐号、数据库用户，并授予其访问学生数据库的读权限；

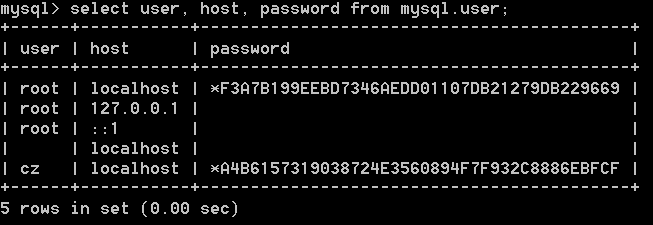






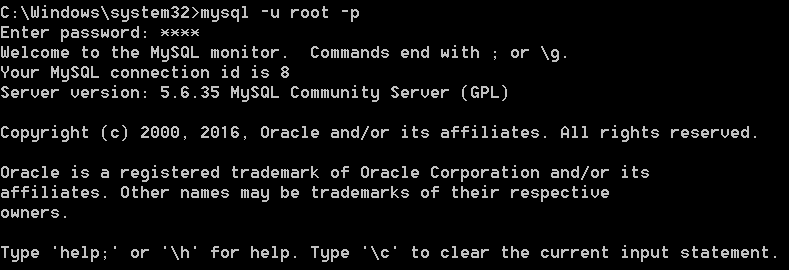
对cz授予访问student的读权限。



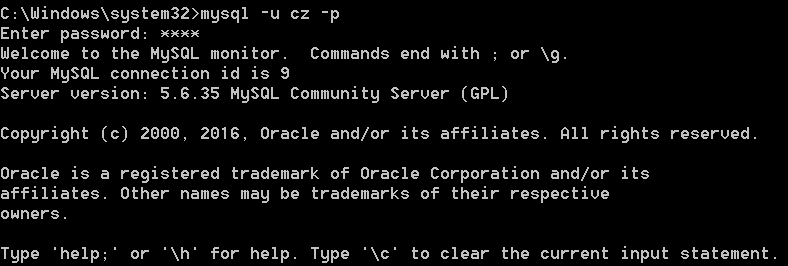


（2） 分别用超级用户和新定义的用户访问学生数据库，并对其中的学生表数据进行修改；

超级用户登录：



新用户cz登录：



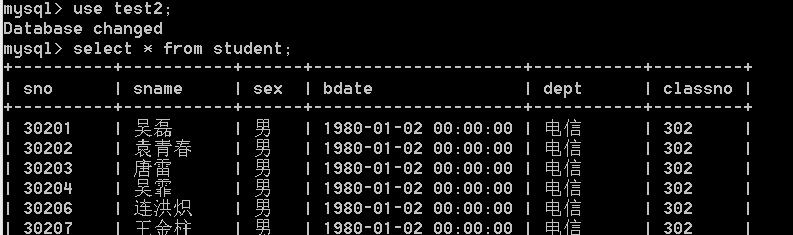
（3） 再次用此用户访问学生数据库，并对其中的学生表数据进行修改。

A、查看student

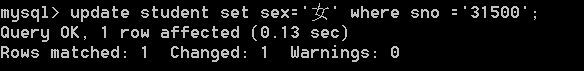
用root查看student



用cz查看student



B、Update一条记录



用户cz update



五、实验结果分析

1、实体完整性约束，主键能够标识表中每条信息的唯一性，不可以重复。因此，在插入与已有元组主键相同的元组时，会报错。

2、参照完整性约束，如果插入的元组的外键不在被参照的表里，那么会报错。

3、定义存储过程，调用后，能成功执行调用相应过程，实现查找

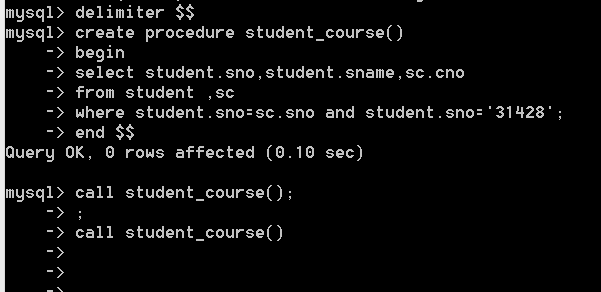
4、建立触发器后，相应可以被触发，更新表项

5、添加新用户后，该用户只能执行授予权限内的操作，不能执行没有授予权限的操作。

**六、实验小结**

（一）问题和解决

1、没有理解delimiter的意思，在定义存储过程后，调用时，没有写终止符。delimiter是分割符的意思，因为MySQL默认以";"为分隔符。声明分割符，防止把存储过程当成SQL语句进行处理，存储过程的编译过程会报错。



2、在建立触发器时，出现错误



MySQL对触发器的支持还没有像SQL Server那么充分，不能再原表上操作，需要创建新表来进行触发操作。

（二）心得

这次实验，我更加深入地理解了实体完整性约束和外键的参照完整性约束，如何定义存储过程，如何建立触发器，如何添加新用户以及更改他的操作权限等sql语句。尽管在实验过程有一些问题，如上述，但都通过查阅资料得以解决。这次实验不仅让我在实践中巩固了课堂所学信息到的理论，并且切身体会到数据库的完整性和安全性设计。这加深了我对数据库学习的兴趣。