# Database System Lab 3

## 1.实验目的

1.熟悉通过SQL进行数据完整性控制的方法；

2.了解数据库在完整性控制下的插入删除限制；

3.掌握触发器的设计与应用。

## 2.实验内容

1.定义若干表，其中包括primary key，foreign key和check的定义；

2.向表中插入数据，考察primary key如何控制实体完整性；

3.删除被引用表中的行，考察foreign key中on delete子句如何控制参照完整性（注意不同类型的功能）；

4.修改被引用表中的行的primary key，考察foreign key中on update子句如何控制参照完整性；

5.修改或插入表中数据，考察check子句如何控制校验完整性；

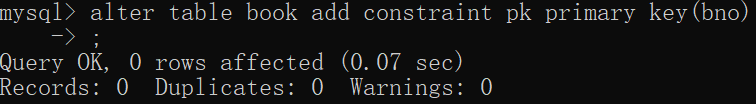
6.定义一个assertion，并通过修改表中数据考察断言如何控制数据完整性；

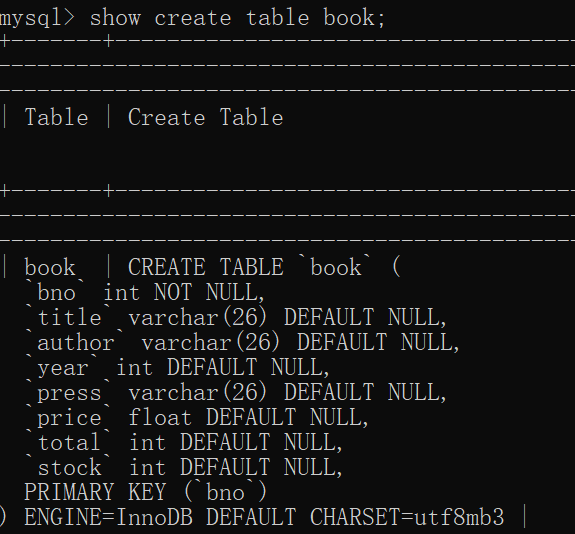
7.定义一个trigger，并通过修改表中数据考察触发器如何起作用。

## 3.实验步骤

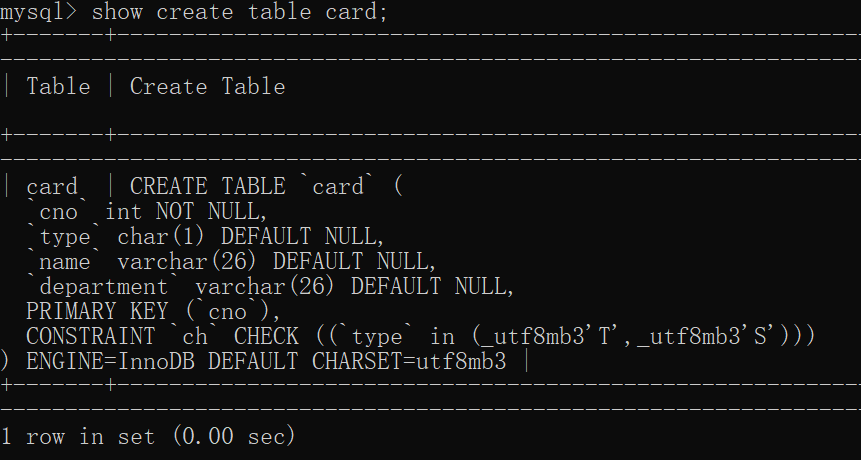
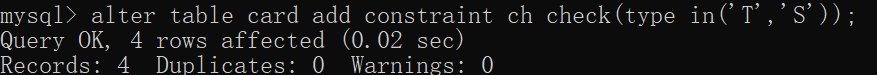
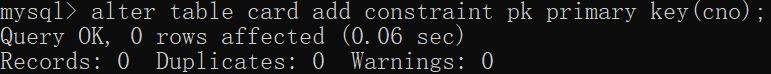
1.在lab2所创立的表的基础上，使用alter table语句增加以下约束条件：

1.1 bno作为book表的主键，代码输入如下：

使用 show create table book语句查看语句结果（见下图），我们发现，主键的限制成功附加；



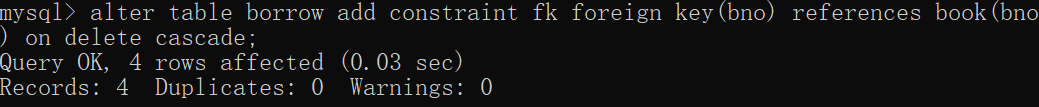
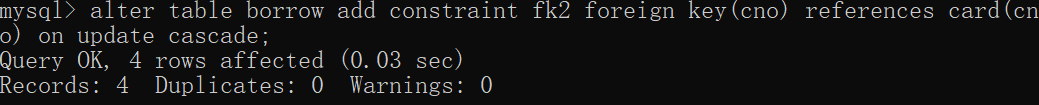
1.2 cno约束为card表的主键，处理方法同1.1的基础上，对card 表中type类型的取值进行范围约束限定，具体实现代码如下：



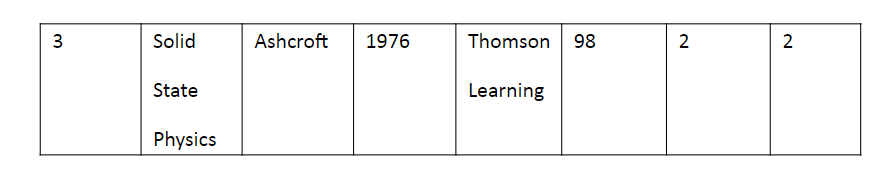
1.3 在borrow表中，将属性值bno作为外键参照book表，且当book表中的记录被删除时，同时删除borrow表中对应的记录（级联删除）。

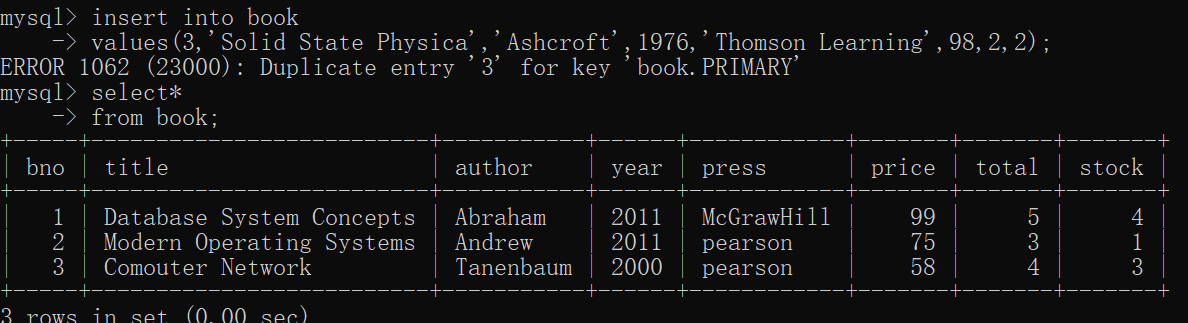
cno作为外键参照card表，且当card表中的记录更新时，同时更新borrow表中对应的记录（级联更新）。

附加外键约束alter table 语句见下：



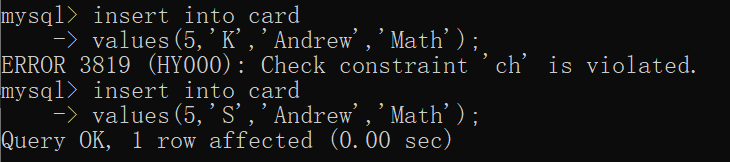
通过 show create table borrow 语句查看结果，同理可得，外键信息已被附加到borrow表中的bno、cno属性上。

1. 向book表中插入指定信息（各属性值如下图），观察反馈结果（见下），反馈结果如下：



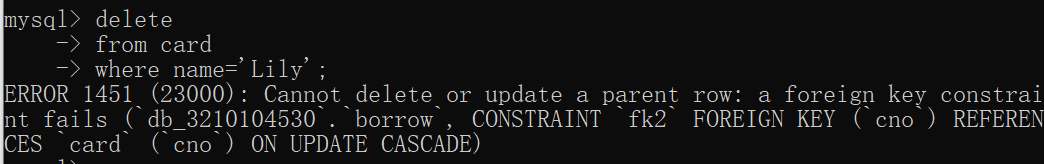
结合select语句得到的book表输出可见，因为我们设置book表中bno属性为该表的主码，因而在尝试插入新数据时，系统会自动判断插入主码是否与已有主码重复，若重复，则会触发error，不允许此次插入。

**3.**向card表插入以下指定信息（各属性值如下图），观察反馈结果（见下），反馈结果如下：

card插入

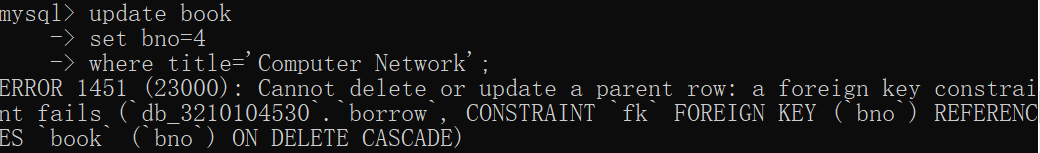
由上两图可知，插入错误；这是因为在crad表的type属性上，我们附加了约束以框定其取值范围（‘T’/‘S’），因而当输入为’K’不属于该范围时，系统不允许此次插入，因而插入失败。

**4.**尝试向card表中删除Lily的借书证，输入代码及得到结果如下：



由error可知，因为在borrow中有外键约束的存在，且只使用了 on update cascade语句进行更新级联因素；若只删除了card表中的Lily借书卡，相应的cno主键也会消失，从而在borrow中没有相应的对映，会引发错误，从而系统不允许这次删除。

1. 尝试修改book表中title为Computer Network书籍的bno为4，输入代码及得到结果如下：



由error结果可以看出，此次更新失败；这也是因为在borrow中对用bno属性有外键约束的影响，在创立此外键约束时，我们只使用了 on delete cascade的级联语句，并未考虑更新的级联，因而book表中bno的更新会导致borrow表中相应bno属性值没有外键对应，从而引发错误，因而更新被拒绝；如果我们希望能成功更新该字段，可以在对borrow表的bno属性进行外键约束时，加上on delete cascade 与 on update cascade语句，这样就可以实现级联更新。

1. 要求：创建一个触发器，使得当向borrow表中插入一条记录时，先检查book表中该书是否有库存：如果库存不为0，则成功借出，并将book表中该书库存减1；如果库存为0，则拒绝借出操作。

触发器设计代码如下：

***//重定义结束符***

delimiter $

create trigger ABC

***//在borrow语句插入前执行触发器***

before insert on borrow

for each row

begin

***//从book表中寻得请求借阅书籍的库存值***

declare re int;

select stock into re

from book

where book.bno=new.bno;

***//判断库存值数量，若大于0，则库存值-1；若小于0，则返回借阅错误信息***

if re>0 then

update book

set book.stock=book.stock-1

where book.bno=new.bno;

else signal sqlstate 'HY000' set message\_text = '已借完';

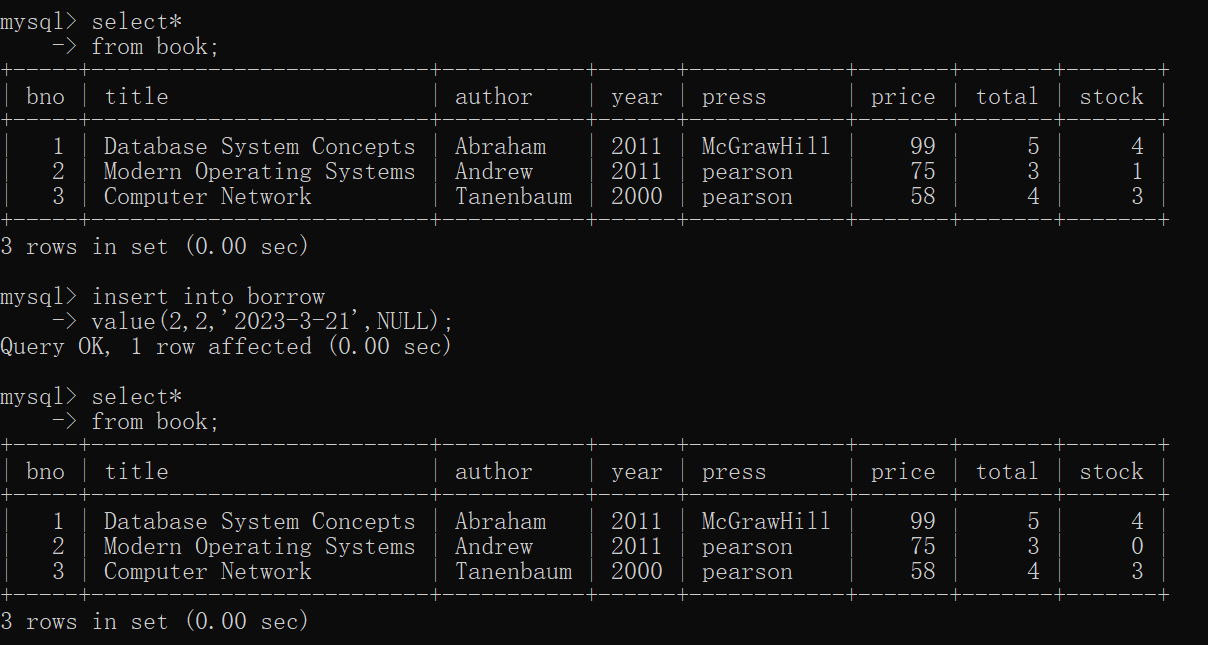
end if;

end $

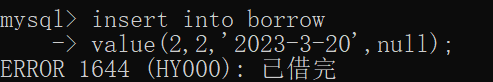
***//更改回结束符定义***

delimiter ;

现在，我们尝试从图书馆借阅一本书---向borrow表插入一条数据，数据及运行结果如下（使用select语句反映book表的变化）：



由输出结果对比后可见，我们在向图书馆借阅一本title 为‘Modern Operating Systems’的书后，该书的库存从1转变为0，符合触发器对于库存大于0的操作；接着，我们继续尝试借出此书，得到的结果如下：



得到上述结果，符合我们触发器的设计，因为该书的库存已为0，因而在此尝试借阅时，需要返回“已借完”的信息，说明触发器模块设计的正确性。

## 4.实验感想

在经历了三个lab之后，我对于SQL语言的特性也有了越来越清晰的认识；但在这三次lab的实践过程中，我也发现了共性的问题，就是对SQL语句何时使用分号的理解（还要稍微吐槽一下mySQL笨笨的操作模式---跨行不能重修改，执行语句不能撤回等等等等）；虽然在一般的语句中，我已明晰了分号的使用，但在触发器的设计中，这一概念又变得模糊---我并不知道怎样算一条完整的语句，而那些提示词语句不是。

当然，导致这个问题的原因，也可能是对于触发器设计的不熟悉，虽然在上网学习，查询一些资料后，我完成了此次任务的设计，但我总感觉有依样画葫芦之感，自己对于一些触发器设计的内涵并没有得到深刻的理解，期待下节课的理论学习可以解开这些谜团。