**3 SQL数据完整性**

**实验目的：**

1. 熟悉通过SQL进行数据完整性控制的方法。

**实验平台：**

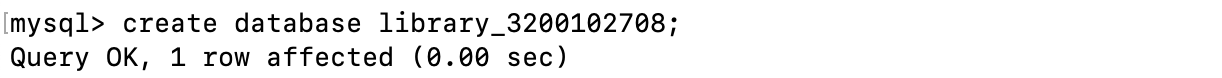
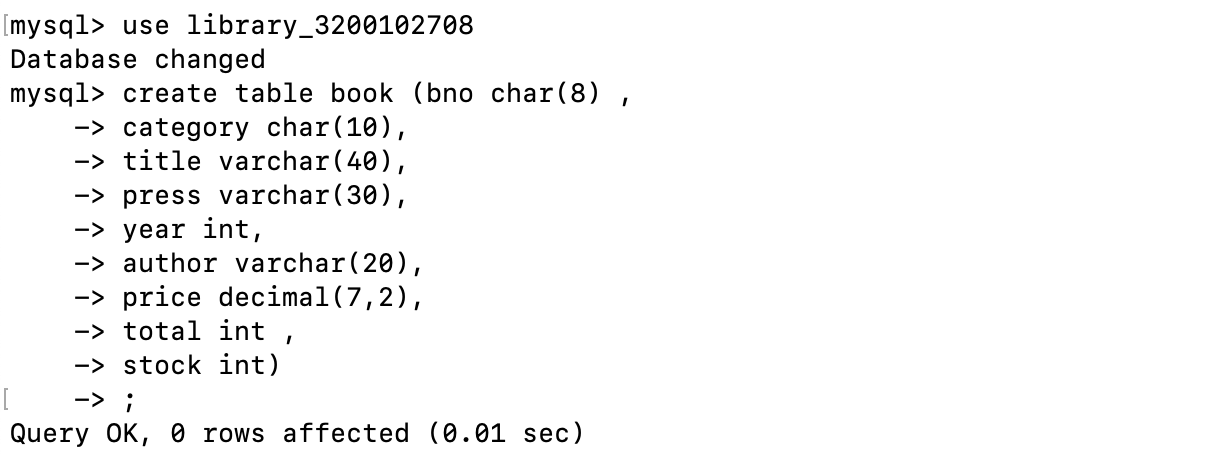
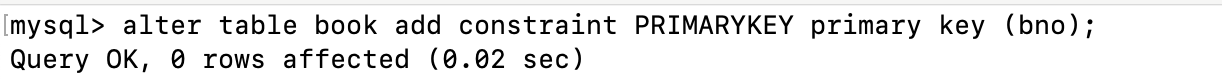
1. 数据库管理系统：mySQL或其它。

**实验内容和要求：**

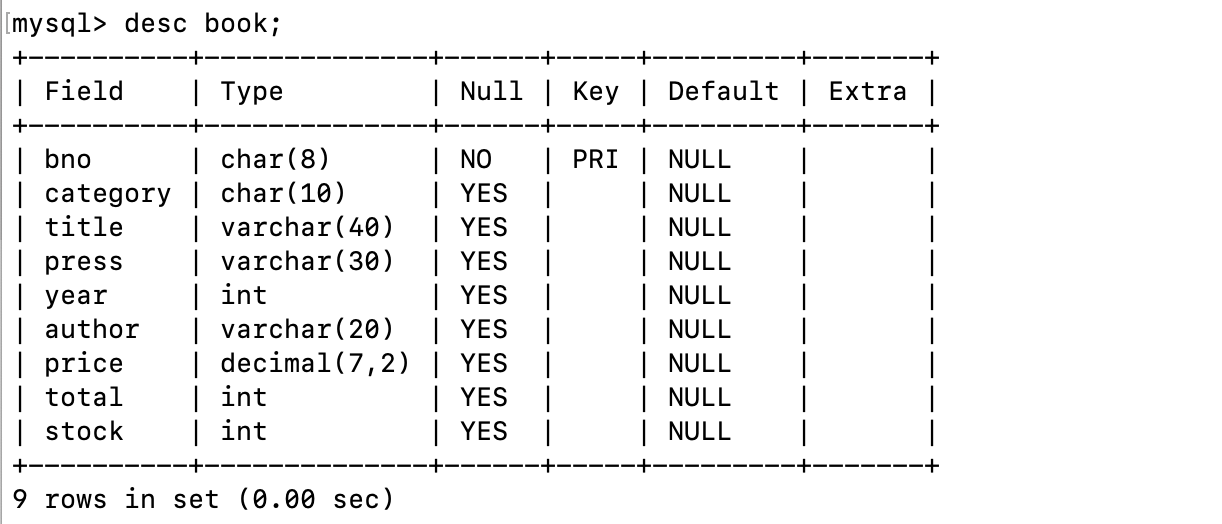
1. 定义若干表，其中包括primary key, foreign key 和check的定义。
2. 让表中插入数据，考察primary key如何控制实体完整性。
3. 删除被引用表中的行，考察foreign key 中on delete 子句如何控制参照完整性。
4. 修改被引用表中的行的primary key，考察foreign key 中on update 子句如何控制参照完整性。
5. 修改或插入表中数据，考察check子句如何控制校验完整性。
6. 定义一个trigger, 并通过修改表中数据考察触发器如何起作用。

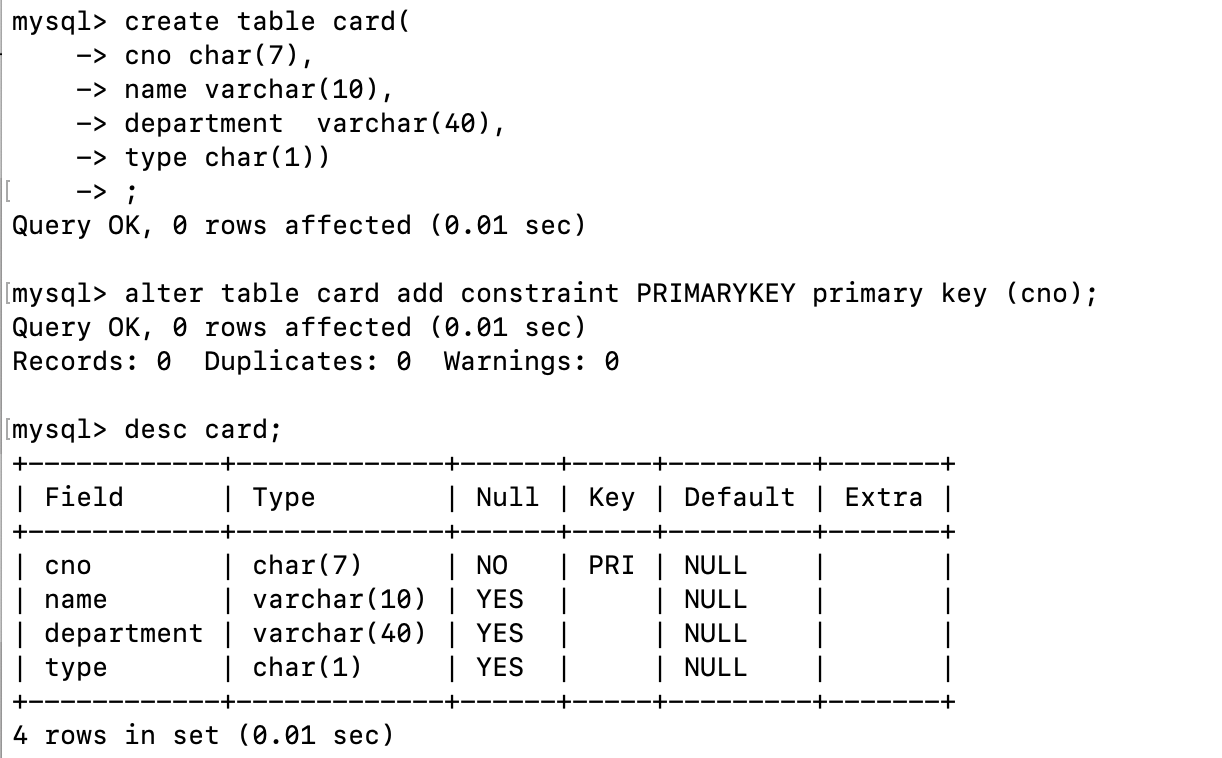
**实验三过程和实验结果：**

**（由学生补充完成，并加上每步实验的执行的sql语句和执行结果的截图）**

1. **创建library数据库，在数据库中创建数据表**

|  |
| --- |
| book (bno char(8) ,  category char(10),  title varchar(40),  press varchar(30),  year int,  author varchar(20),  price decimal(7,2),  total int ,  stock int)  主键：bno |

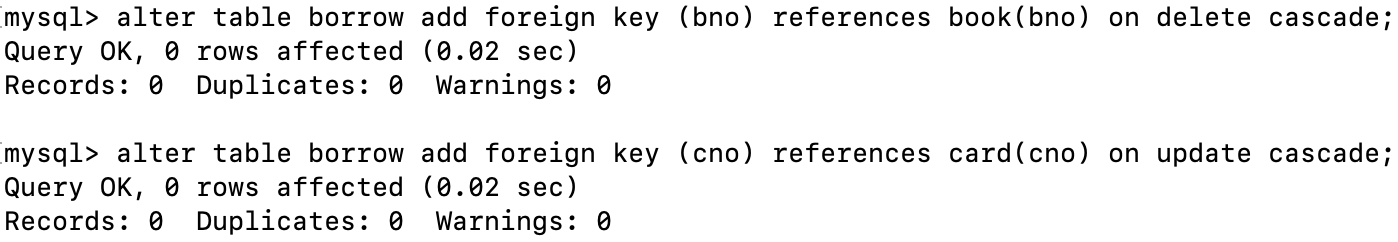
****

1. **编写创建数据表**

|  |
| --- |
| card(  cno char(7),  name varchar(10),  department varchar(40),  type char(1))  主键：cno  type字段 in ('T','G','U','O')；  提示：type字段可以定义为enum 类型 |

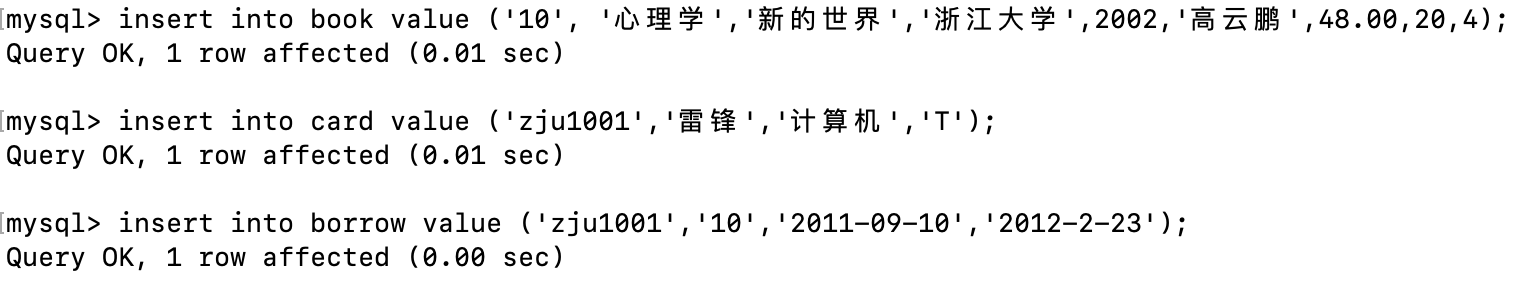
1. **编写创建数据表**

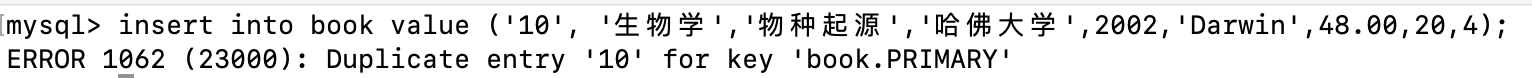
|  |
| --- |
| borrow(  cno char(7),  bno char(8),  borrow\_date datetime,  return\_date datetime,)  外键：bno，cno。 分别对应book和card的bno和cno，同时外键 bno 级联删除 ，cno 级联更新 |



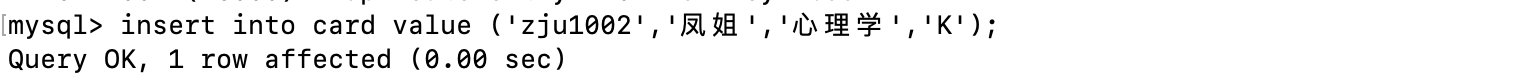
1. **插入基本数据**

|  |
| --- |
| Book表('10', '心理学','新的世界','浙江大学',2002,'高云鹏',48.00,20,4)  Card表('zju1001','雷锋','计算机','T');  Borrow表('zju1001','10','2011-09-10','2012-2-23'); |

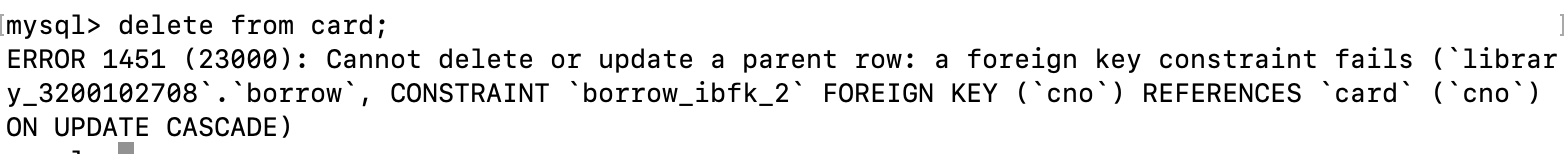
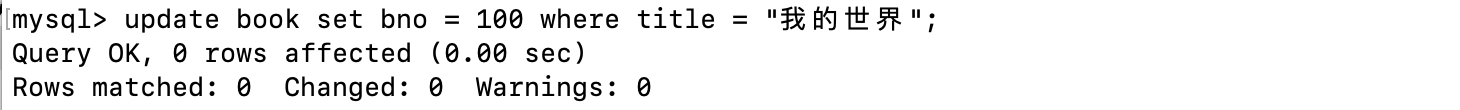


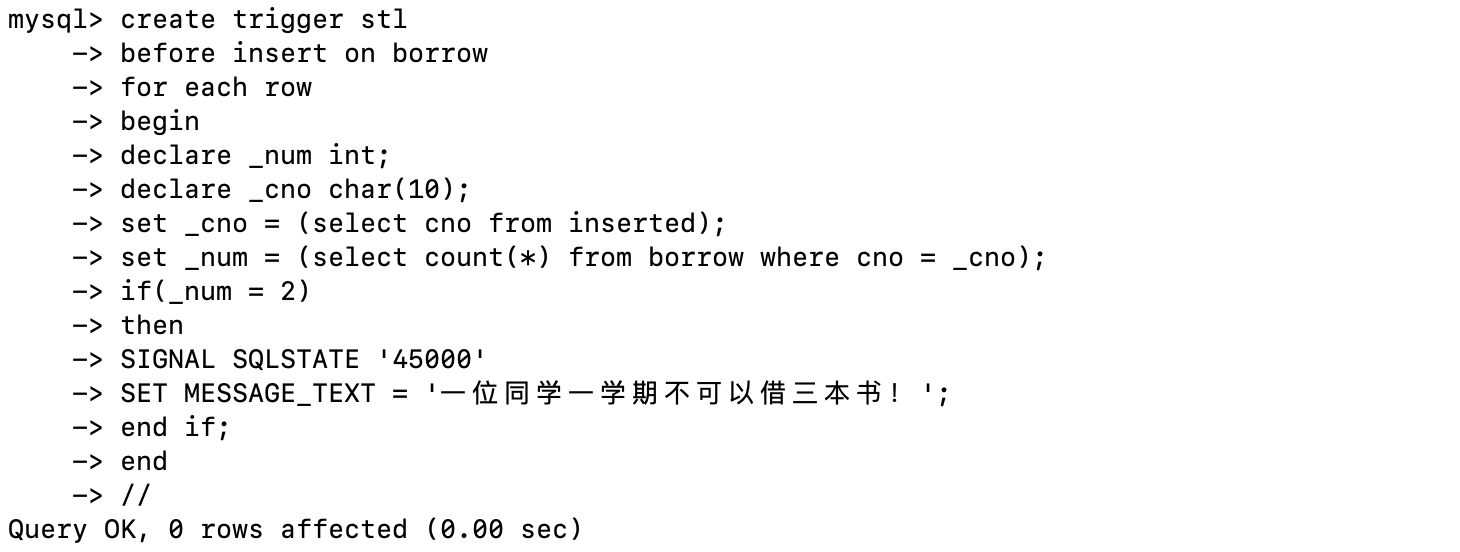
1. **向book表，再次插入一条记录， 看看数据库的提示。**

|  |
| --- |
| ('10', '生物学','物种起源','哈佛大学',2002,'Darwin',48.00,20,4) |

1. **向card表添加一条记录，看看数据库的提示。**

|  |
| --- |
| ('zju1002','凤姐','心理学','K') |

1. **删除card表中的记录，观察数据库的提示。**
2. **更新Book表中‘新的世界’bno 为‘100’看看borrow表中的记录**
3. **编写触发器**

**** 作用是一个借书证不能同时借三本书（简化一点，认为借书记录中有记录即为借了书）。大家尝试编写下先验（before）触发器，作为对比，下面是后验触发器的例子（例子为sql server语法，同学们的答案可以是mysql）。

|  |
| --- |
| create trigger st1 on borrow  after insert  as  Begin  Declare @num int  Declare @cno char(10)  select @cno = cno from inserted  select @num=count(\*) from borrow where cno=@cno  If(@num>2)  begin  Print('一位同学一个学期不可以借三本书！')  Rollback  End  End |

1. **实验总结及思考**

本次实验内容量不大，主要的难点在于对SQL的语法和开发环境不够熟悉，加之上课只会泛泛的讲SQL共通的特性，落实到各个具体的BDSM时又各有区别，我花费了相当多的时间进行摸索。

同时当数据库相关的代码量逐渐变大，业务逻辑逐渐复杂，我开始意识到纯粹命令行操作的不便之处，现在已经开始摸索图形化的数据库IDE。