

数据库系统实验报告

课程名称：____数据库系统____实验类型：____上机____

实验项目名称：____SQL 数据完整性____

学生姓名：____应承峻____专业：____软件工程____学号：____3170103456____

实验日期：____2019____年____3____月____27____日

一、实验目的：

- 熟悉通过 SQL 进行数据完整性控制的方法。

二、实验平台：

- 数据库管理系统：SQL Server 2008 或者其它数据库

三、实验内容和要求：

- 定义若干表，其中包括 primary key, foreign key 和 check 的定义。
- 让表中插入数据，考察 primary key 如何控制实体完整性。
- 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。
- 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。
- 修改或插入表中数据，考察 check 子句如何控制校验完整性。
- 定义一个 trigger，并通过修改表中数据考察触发器如何起作用。

四、实验过程和结果：

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. **create database** library;
- 2.
3. **use library;**

```

4.
5.  create table book(
6.      bno char(8) ,
7.      category char(10),
8.      title varchar(40),
9.      press varchar(30),
10.     year int,
11.     author varchar(20),
12.     price decimal(7,2),
13.     total int,
14.     stock int,
15.     primary key (bno)
16. ) DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
17.
18. desc book;

```

```

mysql> create database library;
Query OK, 1 row affected (0.00 sec)

mysql> use library;
Database changed
mysql> create table 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,
-> primary key (bno)
-> );
Query OK, 0 rows affected (0.26 sec)

```

查看表的结构以验证建表正确性：

```

mysql> desc book;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| bno   | char(8) | NO | PRI | NULL |  |
| category | char(10) | YES |  | NULL |  |
| title | varchar(40) | YES |  | NULL |  |
| press | varchar(30) | YES |  | NULL |  |
| year  | int(11) | YES |  | NULL |  |
| author | varchar(20) | YES |  | NULL |  |
| price | decimal(7,2) | YES |  | NULL |  |
| total | int(11) | YES |  | NULL |  |
| stock | int(11) | YES |  | NULL |  |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.01 sec)

```

2. 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')

```
1. create table card(  
2.     cno char(7),  
3.     name varchar(10),  
4.     department varchar(40),  
5.     type char(1),  
6.     primary key (cno),  
7.     check (type in ('T','G','U','O'))  
8. ) DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
```

```
mysql> create table card(  
-> cno char(7),  
-> name varchar(10),  
-> department varchar(40),  
-> type char(1),  
-> primary key (cno),  
-> check (type in ('T','G','U','O'))  
-> );  
Query OK, 0 rows affected (0.23 sec)
```

查看表的结构以验证建表正确性:

```
mysql> desc card;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| cno        | char(7)       | NO   | PRI | NULL    |       |  
| name       | varchar(10)   | YES  |     | NULL    |       |  
| department | varchar(40)   | YES  |     | NULL    |       |  
| type       | char(1)       | YES  |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)
```

3. 编写创建数据表

borrow(

cno char(7),
bno char(8),
borrow_date datetime,

return_date datetime,)

外键: **bn**o, **cn**o。 分别对应 **book** 和 **card** 的 **bn**o 和 **cn**o, 同时外键 **bn**o 级联删除, **cn**o 级联更新

```
1. create table borrow(  
2.     cno char(7),  
3.     bno char(8),  
4.     borrow_date datetime,  
5.     return_date datetime,  
6.     foreign key (bno) references book(bno) on delete cascade,  
7.     foreign key (cno) references card(cno) on update cascade  
8. ) DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
```

```
mysql> create table borrow(  
->     cno char(7),  
->     bno char(8),  
->     borrow_date datetime,  
->     return_date datetime,  
->     foreign key (bno) references book(bno) on delete cascade,  
->     foreign key (cno) references card(cno) on update cascade  
-> );  
Query OK, 0 rows affected (0.28 sec)
```

查看表的结构以验证建表正确性:

```
mysql> desc borrow;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type      | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| cno        | char(7)   | YES  | MUL | NULL    |       |  
| bno        | char(8)   | YES  | MUL | NULL    |       |  
| borrow_date | datetime  | YES  |     | NULL    |       |  
| return_date | datetime  | YES  |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)
```

4. 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. insert into book values('10', '心理学', '新的世界', '浙江大学', 2002, '高云  
   鹏', 48.00, 20, 4);  
2.  
3. insert into card values('zju1001', '雷锋', '计算机', 'T');  
4.  
5. insert into borrow values('zju1001', '10', '2011-09-10', '2012-2-23');
```

```
mysql> insert into book values('10', '心理学', '新的世界', '浙江大学', 2002, '高云鹏', 48.00, 20, 4);
Query OK, 1 row affected (0.19 sec)

mysql>
mysql> insert into card values('zju1001', '雷锋', '计算机', 'T');
Query OK, 1 row affected (0.03 sec)

mysql>
mysql> insert into borrow values('zju1001', '10', '2011-09-10', '2012-2-23');
Query OK, 1 row affected (0.03 sec)
```

查看表的内容以验证正确性:

```
mysql> select * from book;
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| bno | category | title | press | year | author | price | total | stock |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| 10 | 心理学 | 新的世界 | 浙江大学 | 2002 | 高云鹏 | 48.00 | 20 | 4 |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from card;
+----+-----+-----+-----+
| cno | name | department | type |
+----+-----+-----+-----+
| zju1001 | 雷锋 | 计算机 | T |
+----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from borrow;
+----+-----+-----+-----+
| cno | bno | borrow_date | return_date |
+----+-----+-----+-----+
| zju1001 | 10 | 2011-09-10 00:00:00 | 2012-02-23 00:00:00 |
+----+-----+-----+-----+
1 row in set (0.00 sec)
```

5. 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. `insert into book values('10', '生物学', '物种起源', '哈佛大学', 2002, 'Darwin', 48.00, 20, 4);`

```
mysql> insert into book values('10', '生物学', '物种起源', '哈佛大学', 2002, 'Darwin', 48.00, 20, 4);
ERROR 1062 (23000): Duplicate entry '10' for key 'PRIMARY'
```

错误原因: 已经存在 bno 为10的元组, 而 bno 为 primary key, 主码不允许重复, 所以出错。

6. 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. `insert into card values('zju1002', '凤姐', '心理学', 'K');`

```
mysql> insert into card values('zju1002', '凤姐', '心理学', 'K');
Query OK, 1 row affected (0.13 sec)
```

发现竟然成功插入, 与预期结果插入失败(不在允许的字符内)不符, 查询资料得知 mysql 中的 check 并非强制的 check, 因此需要做如下更改将 type 字段变成枚举类型:

1. `alter table card modify type enum('T', 'G', 'U', 'O');`

此时再做插入则会报错, 达到预期效果:

```
mysql> insert into card values('zju1002', '凤姐', '心理学', 'K');
ERROR 1062 (23000): Duplicate entry 'zju1002' for key 'PRIMARY'
```

7. 删除 card 表中的记录，观察数据库的提示。

```
mysql> delete from card;
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (`library`.`borrow`, CONSTRAINT borrow_ibfk_2 FOREIGN KEY (cno) REFERENCES card (cno) ON UPDATE CASCADE)
```

错误原因：由于 **borrow** 表中存在外键约束，且没定义删除时的行为。

8. 更新 Book 表中‘新的世界’ bno 为‘100’看看 borrow 表中的记录

```
mysql> update book set bno='100' where title = '新的世界';
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (`library`.`borrow`, CONSTRAINT borrow_ibfk_1 FOREIGN KEY (bno) REFERENCES book (bno) ON DELETE CASCADE)
```

错误原因：由于 **book** 表中存在外键约束，且没定义更新时的行为。

9. 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（**instead of**）触发器，作为对比，下面是后验触发器的例子。

```
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
```

由于 **mysql** 中没有 **instead of** 触发器，所以我们使用 **before**

```
1. delimiter |
2. CREATE TRIGGER st1 BEFORE INSERT ON borrow FOR EACH ROW
3. BEGIN
4.     DECLARE num int;
5.     DECLARE msgs VARCHAR(60);
6.     SET num = (select count(cno) from borrow where cno = NEW.cno)
7.     ;
8.     IF num>=2 THEN
9.         SELECT 'cannot brorow three books' INTO @msgs;
```

```
10. END |
11.
12. delimiter;
```

在插入第三条数据后，我们执行语句 `select @msgs`，得到：

```
mysql> select @msgs;
+-----+
| @msgs |
+-----+
| cannot borrow three books |
+-----+
1 row in set (0.00 sec)
```

10. 实验总结及思考

本次实验花费了较多的时间来完成，原因是 `mysql` 的一些语法与书上的不同，例如在 `mysql` 中 `check` 不是强制的，因而需要使用枚举 `enum` 来枚举离散型变量。除此之外还有外键约束，书上的语法是：

foreign key (key_name) references tablename

而在 `mysql` 中，将会包语法错误，正确的语法为：

foreign key (key_name) references tablename(key_name)

再就是触发器，`mysql` 中只有 `before` 和 `after` 两类触发器，麻烦的是 `before` 触发器无论是否满足触发条件，最终的 `sql` 语句都将被执行，这对触发器的设计产生了很大的影响。