

数据库系统实验报告（三）

课程名称： 数据库系统原理 实验项目名称： SQL 数据完整性

学生姓名： 刘轩铭 专业： 软件工程 学号： 3180106071

指导老师： 周波 实验日期： 2020 年 4 月 2 日

一、实验目的和要求

1. 熟悉通过 SQL 进行数据完整性控制的方法

二、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 asseration, 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。
8. 完成实验报告。

三、主要仪器设备

1. 操作系统： Windows
2. 数据库管理系统： SQL Server 或 MySQL （本次实验选用 MySQL）

四、操作方法与实验步骤

4.1 建立一个 Library 数据库，包含完整性定义

我们先定义如下的图书馆数据库：

```

1. book (
2. bno char(8) ,
3. category char(10),
4. title varchar(40),
5. press varchar(30),
6. year int,
7. author varchar(20),
8. price decimal(7,2),
9. total int,
10. stock int
11. )
12.
13. card(
14. cno char(7),
15. name varchar(10),
16. department varchar(40),
17. type char(1)
18. )
19.
20. borrow(
21. cno char(7),
22. bno char(8),
23. borrow_date datetime,
24. return_date datetime
25. )

```

其中加粗的部分为对应的主键。

然后根据 MySQL 语法进行数据库的建立。其中对于完整性定义，我们定义 book 关系的主键:bno; card 关系的主键:cno，因为 type in ('T','G','U','O')，由于 MySQL 不支持 check，故采用 enum; borrow 关系的外键: 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。

打开 MySQL，输入下列代码：

```

create database lab4_Library;
use lab4_Library;
create table book(
    bno char(8) not null,
    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)
);

create table card(
    cno char(7) not null,
    name varchar(10),
    department varchar(40),
    type enum('T','G','U','O'),
    primary key(cno)
);

create table borrow(
    cno char(7) not null,
    bno char(8) not null,
    borrow_date datetime,
    return_date datetime,
    foreign key(bno) references book(bno) on delete cascade,
    foreign key(cno) references card(cno) on update cascade
) ENGINE = InnoDB DEFAULT CHARSET = utf8;

```

完成了该数据库的建立

4.2 向表中插入数据，考察 primary key 如何控制实体完整性

插入以下几条数据：

```

insert into book values('10', '心理学', '新的世界', '浙江大学', 2002, '高云鹏', 48.00, 20, 4);
insert into card values('zju1001', '雷锋', '计算机', 'T');
insert into borrow values('zju1001', '10', '2019-09-10', '2020-02-23');

```

此时插入均没有问题。考虑主码不能重复的完整性定义，再插入下面的数据：

```

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

```

由于 book 表已经存在 bno 为 ‘10’ 的主键，所以再次插入主键为 ‘10’ 的数据时会发生冲突，要将新插入数据的主键 bno 改为 book 表中不存在的值。如下：

```

insert into book values('11', '生物学', '物种起源', '哈佛大学', 2002, 'Darwin', 48.00, 20, 4);

```

此时插入成功了。

4.3 修改或插入表中数据，考察 check 子句如何控制校验完整性

由于 MySQL 不支持 check 子句，我们此处都用 enum 类型进行代替。

尝试插入如下的一条元组：

```
insert into card values('zju1002','东野圭吾','心理学','K');
```

此时插入失败，显示 Type 属性的值不合法。尝试改变 Type 类型，然后再次插入相同的数据：

```
alter table card change type type enum('T','G','U','O','K');  
insert into card values('zju1002','东野圭吾','心理学','K');
```

此时插入成功了。

4.4 删除一条 card 中的数据，观察结果

删除该条数据：

```
delete from book where bno = '10';
```

观察结果可知，相应的 borrow 中参照 bno= '10' 的数据也被删除。则级联删除的效果可以验证。

4.5 更新一条 book 中的数据，观察结果

先插入一条新数据到 borrow，然后修改 card：

```
insert into borrow values('zju1002','11','2019-08-10','2020-03-23');  
update card set cno = 'zju1003' where name = '东野圭吾';
```

观察结果可以知道，相应 borrow 中的数据也被修改，说明级联修改效果成功。

4.6 编写触发器，检查一个借书证不能同时借三本书。

先插入三本新书和一张借书证的新数据：

```
insert into book values('12','计算机','深入理解计算机系统','浙江大学',2002,'Darwin',48.00,20,4);  
insert into book values('13','计算机','算法导论','浙江大学',2002,'Darwin',48.00,20,4);  
insert into book values('14','计算机','数据库系统原理','浙江大学',2002,'Darwin',48.00,20,4);  
insert into card values('zju1010','刘德华','计算机','K');
```

然后编写下面的触发器：

```
delimiter ||
create trigger no_more_than_three after insert on borrow for each row
begin
    if exists (select count(bno) from borrow group by cno having borrow.cno = NEW.cno
and count(bno) > 2)
        then delete from borrow where bno = NEW.bno and cno = NEW.cno;
    end if;
end ||
delimiter;
```

接下来连续插入三条借书数据，借上面三本书，发现第三次借书时出现错误。这说明触发器起到了作用。

```
insert into borrow values('zju1010', '12','2011-09-10','2012-2-23');
insert into borrow values('zju1010', '13','2011-09-10','2012-2-23');
insert into borrow values('zju1010', '14','2011-09-10','2012-2-23');
```

4.6 定义一个断言，并通过修改表中数据考察断言如何控制数据完整性。

由于 MySQL 不支持断言，所以该部分操作不进行。这部分断言的操作可以通过触发器进行完成和代替。

五、实验结果与分析

5.1 建立数据库结果如图

```
mysql> create database lab4_library
-> ;
Query OK, 1 row affected (0.69 sec)

mysql> use lab4_library;
Database changed
mysql> create table book(
-> bno char(8) not null,
-> 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 (0.86 sec)

mysql>
mysql> create table card(
-> cno char(7) not null,
-> name varchar(10),
-> department varchar(40),
-> type enum('T','G','U','O'),
-> primary key(cno)
-> );
Query OK, 0 rows affected (0.22 sec)

mysql>
mysql> create table borrow(
-> cno char(7) not null,
-> bno char(8) not null,
-> borrow_date datetime,
-> return_date datetime,
-> foreign key(bno) references book(bno) on delete cascade,
-> foreign key(cno) references card(cno) on update cascade
-> ) ENGINE = InnoDB DEFAULT CHARSET = utf8;
Query OK, 0 rows affected, 1 warning (0.39 sec)
```

返回结果说明已经建立了该数据库和相应关系。

5.2 插入结果成功，完整性约束导致主码不能相同

```
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 (1.12 sec)

mysql> insert into card values('zjul001', '雷锋', '计算机', 'T');
Query OK, 1 row affected (0.07 sec)

mysql> insert into borrow values('zjul001', '10', '2019-09-10', '2020-02-23');
Query OK, 1 row affected (0.28 sec)

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'
mysql> insert into book values('11', '生物学', '物种起源', '哈佛大学', 2002, 'Darwin', 48.00, 20, 4);
Query OK, 1 row affected (0.15 sec)
```

如图所示，由于 bno= '10' 重复导致数据无法插入，但修改后，插入成功。

5.3 type 值不合法违反了“check”子句的约束，修改后成功

```
mysql> insert into card values('zjul002', '东野圭吾', '心理学', 'K');
ERROR 1265 (01000): Data truncated for column 'type' at row 1
mysql> alter table card change type type enum('T', 'G', 'U', 'O', 'K');
Query OK, 0 rows affected (0.72 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> insert into card values('zjul002', '东野圭吾', '心理学', 'K');
Query OK, 1 row affected (0.04 sec)

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
11	生物学	物种起源	哈佛大学	2002	Darwin	48.00	20	4

2 rows in set (0.13 sec)

如图所示，在修改了 type 的定义后插入成功了。显示 book 中的数据符合预期要求。

5.4 级联删除和级联更新结果：

```
mysql> delete from book where bno = '10';
Query OK, 1 row affected (0.23 sec)

mysql> select * from book
-> ;
```

bno	category	title	press	year	author	price	total	stock
11	生物学	物种起源	哈佛大学	2002	Darwin	48.00	20	4

1 row in set (0.00 sec)

```
mysql> select * from borrow;
Empty set (0.02 sec)
```


可以看到由于 on delete cascading 的约束，删除 book 中数据后，borrow 中的数据也相应被删除，导致 borrow 为空集。

```
mysql> insert into borrow values('zjul002', '11', '2019-08-10', '2020-03-23');
Query OK, 1 row affected (0.16 sec)

mysql> update card set cno = 'zjul003' where name = '东野圭吾';
Query OK, 1 row affected (0.11 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from card
-> ;
+----+-----+-----+-----+
| cno | name  | department | type |
+----+-----+-----+-----+
| zjul001 | 雷锋 | 计算机 | I |
| zjul003 | 东野圭吾 | 心理学 | K |
+----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from borrow:
+----+----+-----+-----+
| cno | bno | borrow_date | return_date |
+----+----+-----+-----+
| zjul003 | 11 | 2019-08-10 00:00:00 | 2020-03-23 00:00:00 |
+----+----+-----+-----+
1 row in set (0.00 sec)
```

可以看到由于 on update cascading 的约束，更新 card 中数据后，borrow 中的数据也相应被更新。

以上结果说明级联删除和级联更新的效果成功了。

5.5 触发器效果展示：

```
mysql> insert into book values('12', '计算机', '深入理解计算机系统', '浙江大学', 2002, 'Darwin', 48.00, 20, 4);
Query OK, 1 row affected (0.51 sec)

mysql> insert into book values('13', '计算机', '算法导论', '浙江大学', 2002, 'Darwin', 48.00, 20, 4);
Query OK, 1 row affected (0.05 sec)

mysql> insert into book values('14', '计算机', '数据库系统原理', '浙江大学', 2002, 'Darwin', 48.00, 20, 4);
Query OK, 1 row affected (0.09 sec)

mysql> insert into card values('zjul010', '刘德华', '计算机', 'K');
Query OK, 1 row affected (0.60 sec)

mysql> delimiter ||
mysql> create trigger no_more_than_three after insert on borrow for each row
-> begin
-> if exists (select count(bno) from borrow group by cno having borrow.cno = NEW.cno and count(bno) > 2)
-> then delete from borrow where bno = NEW.bno and cno = NEW.cno;
-> end if;
-> end ||
Query OK, 0 rows affected (0.40 sec)
```

以上部分插入了需要的数据，并编写了触发器。

```
mysql> insert into borrow values('zjul010', '12', '2011-09-10', '2012-3-23');
-> insert into borrow values('zjul010', '13', '2011-09-10', '2012-3-23');
-> insert into borrow values('zjul010', '14', '2011-09-10', '2012-3-23');
-> ;
Query OK, 1 row affected (0.09 sec)
Query OK, 1 row affected (0.14 sec)
ERROR 1442 (HY000): Can't update table 'borrow' in stored function/trigger because it is already used by statement which invoked this stored function/trigger.
mysql>
```

此时再插入数据，发现第三条插入时出现错误，即触发器生效。

六、讨论与心得

此次实验主要对外键主键的约束以及 trigger 进行了深入的了解。

在此基础上,我查阅了 SQL 的完整性约束的相关资料,主要有以下几个方面:

实体完整性: 使用主键唯一的标识一个实体, 各实体不能相同。

引用完整性: 外键与主键之间的引用规则。在输入或删除记录时, 引用完整性保持表中已定义的关系。比如当为 borrow 中插入一条记录时, 其 cno 和 bno 要首先存在于 card 和 book 中。在 card 或 book 中删除一条记录时, borrow 中对应的 cno 或 bno 要首先删除。这样的一致性的要求不能引用不存在的值, 如果键值改了, 那么整个数据库的键值的所有引用要进行一致的更改。

用户定义完整性: 用户定义不属于其他完整性分类的特定规则, 比如触发器。

MySQL 对基本的完整性约束是支持的, 包括主键约束, 唯一性约束, 外键约束, 非空约束。但 MySQL 不支持 check 约束, 虽然可以使用 check 约束, 但是不起作用。