

北京邮电大学数据库系统实验报告

实验名称: 数据库完整性与安全性实验

计算机科学与技术系

2015211306班

魏晓

学号:2015211301

实验目的

- 通过对完整性规则的定义实现, 熟悉了解kingbase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。
- 通过对安全性相关内容的定义, 熟悉了解kingbase中安全性的内容和实现方法, 加深对数据库安全性的理解

实验内容

- 完整性实验
- 安全性试验

实验平台及环境

实验平台:MySQL 14.14 with Python 3.6

运行环境:Mac OS High Sierra 10.13(17A405)

可视界面:MySQL WorkBench 6.3.10

```
Xiaos-MacBook-Pro:~ weixiao$ mysql --version
mysql Ver 14.14 Distrib 5.7.20, for macos10.12 (x86_64) using EditLine wrapper
```

实验步骤及结果分析

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;

o 先向sc表中添加两个外码约束如下:

```
mysql> alter table sc add constraint foreign key (cno) references course(cno);
Query OK, 142 rows affected (0.04 sec)
Records: 142  Duplicates: 0  Warnings: 0
```

o 然后表的信息如下所示:

```
weixiao — mysql -u wx -p — 80x26
mysql> show create table sc;
+-----+-----+
| Table | Create Table |
+-----+-----+
| sc     | CREATE TABLE `sc` (
  `sno` varchar(6) NOT NULL,
  `cno` varchar(3) NOT NULL,
  `grade` int(11) DEFAULT NULL,
  PRIMARY KEY (`sno`,`cno`),
  KEY `cno` (`cno`),
  CONSTRAINT `sc_ibfk_1` FOREIGN KEY (`cno`) REFERENCES `course` (`cno`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.01 sec)
```

```
weixiao — mysql -u wx -p — 80x26
[mysql> show create table course;
+-----+-----+
| Table | Create Table
+-----+-----+
| course | CREATE TABLE `course` (
  `cno` varchar(3) NOT NULL,
  `cname` varchar(16) DEFAULT NULL,
  `lhour` int(10) DEFAULT NULL,
  `credit` int(10) DEFAULT NULL,
  `semester` varchar(2) DEFAULT NULL,
  PRIMARY KEY (`cno`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.00 sec)
```

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,验证实体完整性约束;

```
[mysql> insert into course values('C01','','90','');
ERROR 1136 (21S01): Column count doesn't match value count at row 1
```

- 可以看到由于插入操作破坏了实体完整性约束,因此插入失败

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;

```
[mysql> insert into sc values('31428','C99',90);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`mytest`.`sc`, CONSTRAINT `sc_ibfk_1` FOREIGN KEY (`cno`) REFERENCES `course` (`cno`))
```

- 这违背了外码约束,其中外码的名称是系统自定义;

4. 删除学生表中所有数据,验证参照完整性约束

```
[mysql> drop table course;
ERROR 1217 (23000): Cannot delete or update a parent row: a foreign key constraint fails
```

- 无法删除这个表,这是因为会破坏sc表的参照性约束

5. 定义存储过程,完成查询某个学生的选课情况,并执行:

```
[mysql> DELIMITER //
[mysql> create procedure show_sc(in sinput varchar(5))
[    -> begin
[    -> select cno
[    -> from sc
[    -> where sno = sinput;
[    -> end
[    -> //
Query OK, 0 rows affected (0.01 sec)
```

- 然后调用这个存储过程,可以看到,他把某一学生的学号作为输入,输出这个学生的选课情况.

```
[mysql> call show_sc('31426');//
+-----+
| cno |
+-----+
| C01 |
| C02 |
| C03 |
+-----+
3 rows in set (0.01 sec)
```

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

```
[mysql> create trigger adate after insert on sc for each row begin update student
set birthdate = DATE_ADD(birthdate,INTERVAL 1 DAY); end //
Query OK, 0 rows affected (0.04 sec)
```

- 然后验证触发器:

```
[mysql> insert into sc values('31428','C04',100);
[    -> //
Query OK, 1 row affected (0.01 sec)
```

- 得到结果:

```
[mysql> select * from student;
--> //]
+-----+-----+-----+-----+-----+-----+-----+
--+
| sno   | sname   | sex   | birthdate           | dept   | classno | nation
|
+-----+-----+-----+-----+-----+-----+-----+
---+
| 30201 | 吴磊     | 男    | 1980-01-03 00:00:00 | 电信   | 302     | zhonggu
p |
| 30202 | 袁青春   | 男    | 1980-01-03 00:00:00 | 电信   | 302     | NULL
|
| 30203 | 唐雷     | 男    | 1980-01-03 00:00:00 | 电信   | 302     | NULL
|
| 30204 | 吴霏     | 男    | 1980-01-03 00:00:00 | 电信   | 302     | NULL
|
| 30206 | 连洪炽   | 男    | 1980-01-03 00:00:00 | 电信   | 302     | NULL
|
| 30207 | 王金柱   | 男    | 1980-01-03 00:00:00 | 电信   | 302     | NULL
|
| 30208 | 苏广学   | 男    | 1980-01-03 00:00:00 | 电信   | 302     | NULL
|
| 30209 | 唐元亮   | 男    | 1980-01-03 00:00:00 | 电信   | 302     | NULL
|
| 30210 | 葛艳杰   | 男    | 1980-01-03 00:00:00 | 电信   | 302     | NULL
|
```

- 可以看到相比之前每个人的出生日期的确增加了一天

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；

- 创建了用户“sa”,只有读权限;

```
mysql> grant select on mytest to 'sa'@'localhost' identified by 'abc123';
```

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


```
mysql> select * from course;
```

cno	cname	lhour	credit	semester
C01	编译原理	51	3	球
C02	数据库原理	51	3	春
C03	操作系统	51	2	秋
C04	JAVA 程序设计	40	2	秋
C05	计算机组成原理	30	2	春
C06	数据结构	60	3	春
C07	math	0	0	sp

7 rows in set (0.00 sec)

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。

```
mysql> insert into course values('C08','english',1,1,'su');  
ERROR 1142 (42000): INSERT command denied to user 'sa'@'localhost' for table 'course'
```

实验总结

- 本实验重点是数据库的完整性约束和安全性约束,这些约束和机制隐匿于平时的操作之下,这位数据库的安全和稳定提供了保证.
- 如何理解完整性约束和安全性约束?
 - 完整性约束可以理解为,当你改变数据库中某一个值的时候,这个值可能是另一个表的外键,或者承担着其他的作用,所以你删除之后就会让另一个数据库缺少键,这些就是完整性约束
 - 数据库在安全运行的时候,会给不同用户不同的权限,不同的权限对应不同的操作,你不能干超过你权限的操作,这种机制就叫安全性约束