

**北京邮电大学 计算机学院**  
**《数据库系统原理》实验报告**

**姓名** 王睿嘉

**学号** 2015211906

**班级** 2015211307

## 实验五 数据库完整性与安全性

### 一、实验内容 and 环境描述

#### 1. 实验目的

- 1) 通过对完整性规则的定义, 熟悉 MySQL 中保证完整性的规则和相应实现方法, 加深对数据完整性的理解;
- 2) 通过对安全性相关内容的定义, 熟悉 MySQL 中保证安全性的实现方法, 加深对数据库安全性的理解。

#### 2. 实验内容

##### 1) 完整性实验

- ① 分别定义学生数据库中各基表的主、外键, 实现实体完整性约束和参照完整性约束;
- ② 分别向学生、课程表插入具有相同学号及课程编号的学生数据和课程数据, 验证其实体完整性约束;
- 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;
- ④ 删除学生表中的所有数据, 验证参照完整性约束;
- ⑤ 定义存储过程, 完成查询某个学生的选课情况, 并执行;
- ⑥ 定义触发器, 当向学生表插入一条新的记录时, 将所有学生出生日期加 1, 并对其进行测试;
- ⑦ 用 SQL 完成以上操作。

##### 2) 安全性实验

- ① 定义新的数据库用户, 并授予其访问学生数据库的读权限;
- ② 分别用 root 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;
- ③ 再次用此用户访问学生数据库, 并对其中的学生数据进行修改;
- ④ 用 SQL 语句分别完成以上内容。

#### 3. 实验环境

数据库系统: 5.7.21 MySQL Community Server (GPL)

### 二、实验结论和心得

#### 1. 实验结论

1.1 分别定义学生数据库中各基表的主、外键, 实现实体完整性约束和参照完整性约束  
在创建表时, 已经定义了各基表的主键。因此, 先把各基表的主键删除, 再添加, 从而实现实体完整性约束:

```

MySQL.exe - mysql -uroot -p mysql
mysql> alter table course drop primary key;
Query OK, 6 rows affected (0.29 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> alter table student drop primary key;
Query OK, 54 rows affected (0.23 sec)
Records: 54 Duplicates: 0 Warnings: 0

mysql> alter table sc drop primary key;
Query OK, 142 rows affected (0.23 sec)
Records: 142 Duplicates: 0 Warnings: 0

mysql> alter table course add primary key(cno);
Query OK, 0 rows affected (0.21 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table student add primary key(sno);
Query OK, 0 rows affected (0.16 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table sc add primary key(cno,sno);
Query OK, 0 rows affected (0.45 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>

```

sc 表中的 cno 和 sno 分别是 course 和 student 表的外关键字。给 sc 表添加相应外键，实现参照完整性约束：

```

MySQL.exe - mysql -uroot -p mysql
mysql> alter table sc add constraint fk_student foreign key(sno) references student(sno);
Query OK, 142 rows affected (0.52 sec)
Records: 142 Duplicates: 0 Warnings: 0

mysql> alter table sc add constraint fk_course foreign key(cno) references course(cno);
Query OK, 142 rows affected (0.15 sec)
Records: 142 Duplicates: 0 Warnings: 0

mysql>

```

主、外键均添加成功后，查看各基表信息：

```

mysql> show create table course;

+-----+
| Table | Create Table |
+-----+
| course | CREATE TABLE `course` (
  cno varchar(3) NOT NULL,
  cname varchar(12) DEFAULT NULL,
  credit int(11) DEFAULT NULL,
  PRIMARY KEY (cno)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+
1 row in set (0.00 sec)

mysql> show create table student;

+-----+
| Table | Create Table |
+-----+
| student | CREATE TABLE `student` (
  sno varchar(6) NOT NULL,
  sname varchar(6) DEFAULT NULL,
  sex varchar(2) DEFAULT NULL,
  bdate datetime DEFAULT NULL,
  dept varchar(8) DEFAULT NULL,
  classno varchar(4) DEFAULT NULL,
  PRIMARY KEY (sno),
  KEY index_sno (sno)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+
1 row in set (0.00 sec)

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 (cno, sno),
  KEY fk_student (sno),
  CONSTRAINT fk_course FOREIGN KEY (cno) REFERENCES `course` (cno),
  CONSTRAINT fk_student FOREIGN KEY (sno) REFERENCES `student` (sno)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+
1 row in set (0.01 sec)

mysql>

```

## 1.2 分别向学生、课程表插入具有相同学号及课程编号的学生和课程数据，验证其实体完整性约束

```
MySQL.exe - mysql -uroot -p mysql
mysql> insert into
  -> course values('C01','软件工程',3);
ERROR 1062 (23000): Duplicate entry 'C01' for key 'PRIMARY'
mysql> insert into
  -> student values('30201','张三','男','1997-01-01 00:00:00','计算机','3017');
ERROR 1062 (23000): Duplicate entry '30201' for key 'PRIMARY'
mysql>
```

分别对 student 和 course 表插入具有相同学号及课程号的学生和课程信息，均提示错误信息: Duplicate entry \*\*\* for key 'PRIMARY'，验证了实体完整性约束。

## 1.3 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.exe - mysql -uroot -p mysql
mysql> insert into
  -> sc values('30201','C07',90);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails ('student registration system','sc', CONSTRAINT 'fk_course' FOREIGN KEY ('cno') REFERENCES 'course' ('cno'))
mysql>
```

学号 '30201' (student 表中存在)，课程号 'C07' (course 表中不存在)，因参照完整性约束，插入失败。

## 1.4 删除学生表中的所有数据，验证参照完整性约束

```
MySQL.exe - mysql -uroot -p mysql
mysql> drop table student;
ERROR 1217 (23000): Cannot delete or update a parent row: a foreign key constraint fails
mysql>
```

sc 表中的 sno 是 student 表的外键，因参照完整性约束，删除失败。

## 1.5 定义存储过程，完成查询某学生的选课情况，并执行

存储过程是一组为了完成特定功能的 SQL 语句集。在数据库中经一次编译后，再调用不需再次编译。

用户通过指定存储过程的名字，并给出相应参数来执行它。

因为在存储过程中会用到结束符“;”，首先用 delimiter 修改结束符为//:

```

MySQL> delimiter //
MySQL> create procedure search_sc(num char(6))
-> begin
-> select sno,cno
-> from sc
-> where sno=num;
-> end//
Query OK, 0 rows affected (0.43 sec)

MySQL> call search_sc('30201')//
+----+----+
| sno | cno |
+----+----+
| 30201 | C03 |
| 30201 | C04 |
| 30201 | C05 |
+----+----+
3 rows in set (0.01 sec)

Query OK, 0 rows affected (0.03 sec)

MySQL>

```

## 1.6 定义触发器，当向学生表中插入一条新的记录时，将所有学生出生日期加 1，并对其进行测试

触发器的功能是监视某种情况，并触发相应操作:

```

MySQL> create table new_student
-> select *
-> from student//
Query OK, 54 rows affected (0.09 sec)
Records: 54 Duplicates: 0 Warnings: 0

MySQL> create trigger t_bdate
-> before insert on student
-> for each row
-> update new_student
-> set bdate=date_add(bdate,interval 1 day)//
Query OK, 0 rows affected (0.04 sec)

MySQL> insert into
-> student values('99999','男','男','1997-01-01 00:00:00','计算机','3007')//
Query OK, 1 row affected (0.05 sec)

MySQL> select *
-> from student//
+----+-----+----+-----+-----+-----+
| sno | sname | sex | bdate | dept | classno |
+----+-----+----+-----+-----+-----+
| 30201 | 吴磊 | 男 | 1989-01-02 00:00:00 | 电信 | 3022 |
| 30202 | 袁青存 | 男 | 1980-01-02 00:00:00 | 电信 | 3022 |
| 30203 | 傅雷 | 男 | 1980-01-03 00:00:00 | 电信 | 3022 |
| 30204 | 吴军 | 男 | 1980-01-02 00:00:00 | 电信 | 3022 |
| 30206 | 连洪斌 | 男 | 1980-01-02 00:00:00 | 电信 | 3022 |
| 30207 | 王金柱 | 男 | 1980-01-02 00:00:00 | 电信 | 3022 |
| 30208 | 苏广学 | 男 | 1980-01-02 00:00:00 | 电信 | 3022 |
| 30209 | 唐元亮 | 男 | 1980-01-02 00:00:00 | 电信 | 3022 |

```

```

MySQL> select *
-> from new_student//
+----+-----+----+-----+-----+-----+
| sno | sname | sex | bdate | dept | classno |
+----+-----+----+-----+-----+-----+
| 30201 | 吴磊 | 男 | 1989-01-03 00:00:00 | 电信 | 3022 |
| 30202 | 袁青存 | 男 | 1980-01-03 00:00:00 | 电信 | 3022 |
| 30203 | 傅雷 | 男 | 1980-01-03 00:00:00 | 电信 | 3022 |
| 30204 | 吴军 | 男 | 1980-01-03 00:00:00 | 电信 | 3022 |
| 30206 | 连洪斌 | 男 | 1980-01-03 00:00:00 | 电信 | 3022 |
| 30207 | 王金柱 | 男 | 1980-01-03 00:00:00 | 电信 | 3022 |
| 30208 | 苏广学 | 男 | 1980-01-03 00:00:00 | 电信 | 3022 |
| 30209 | 唐元亮 | 男 | 1980-01-03 00:00:00 | 电信 | 3022 |
| 30210 | 葛振杰 | 男 | 1980-01-03 00:00:00 | 电信 | 3022 |
| 30211 | 戴永超 | 男 | 1980-01-03 00:00:00 | 电信 | 3022 |
| 30212 | 张伟 | 男 | 1980-01-03 00:00:00 | 电信 | 3022 |
| 30213 | 孙明 | 男 | 1980-01-03 00:00:00 | 电信 | 3022 |
| 30214 | 车平跃 | 男 | 1980-01-03 00:00:00 | 电信 | 3022 |
| 30215 | 张鑫 | 男 | 1980-01-03 00:00:00 | 电信 | 3022 |
| 30216 | 江帆 | 男 | 1980-01-03 00:00:00 | 电信 | 3022 |
| 30217 | 马涛 | 男 | 1980-01-03 00:00:00 | 电信 | 3022 |
| 30218 | 王仕楠 | 男 | 1980-01-03 00:00:00 | 电信 | 3022 |
| 30219 | 马浩 | 男 | 1980-01-03 00:00:00 | 电信 | 3022 |
| 30220 | 邵苏杰 | 男 | 1980-01-03 00:00:00 | 电信 | 3022 |
| 30221 | 邵建伟 | 男 | 1980-01-03 00:00:00 | 电信 | 3022 |
| 30222 | 郑祖彦 | 男 | 1980-01-03 00:00:00 | 电信 | 3022 |
| 30223 | 丁润成 | 女 | 1980-01-03 00:00:00 | 电信 | 3022 |
| 30224 | 赵海燕 | 女 | 1980-01-03 00:00:00 | 电信 | 3022 |
| 30225 | 陈菊英 | 女 | 1980-01-03 00:00:00 | 电信 | 3022 |
| 30226 | 李丽云 | 女 | 1980-01-03 00:00:00 | 电信 | 3022 |

```

更新后: student 表中的数据没有发生改变，只是插入了一条新的学生信息；而 new\_student 表中所有学生的出生日期都增加了 1，但没有新插入的学生信息。

## 2.1 定义新的数据库用户，并授予其访问学生数据库的读权限

```

MySQL> create user 'wrj'@'%' identified by '142857';
Query OK, 0 rows affected (0.37 sec)

MySQL> flush privileges;
Query OK, 0 rows affected (0.09 sec)

MySQL> grant select on student to wrj;
Query OK, 0 rows affected (0.09 sec)

MySQL> grant select on course to wrj;
Query OK, 0 rows affected (0.00 sec)

MySQL> grant select on sc to wrj;
Query OK, 0 rows affected (0.29 sec)

MySQL>

```

## 2.2 分别用 root 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

root 用户：

```
MySQLExe - mysql -u root -p
C:\Windows\System32\mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 5.7.21 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use 'Student Registration System';
Database changed
mysql> select *
    -> from course;
+----+-----+-----+
| cno | cname | credit |
+----+-----+-----+
| C01 | 编译原理 | 3 |
| C02 | 数据库原理 | 3 |
| C03 | 操作系统 | 2 |
| C04 | JAVA 程序设计 | 3 |
| C05 | 计算机组成原理 | 3 |
| C10 | 软件工程 | 3 |
+----+-----+-----+
6 rows in set (0.00 sec)

mysql> delete from
    -> course
    -> where cno='C10';
Query OK, 1 row affected (0.00 sec)

mysql>
```

新定义的用户：

```
MySQLExe - mysql -u wrj -p
C:\Windows\System32\mysql -u wrj -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 5.7.21 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use 'Student Registration System';
Database changed
mysql> select *
    -> from course;
+----+-----+-----+
| cno | cname | credit |
+----+-----+-----+
| C01 | 编译原理 | 3 |
| C02 | 数据库原理 | 3 |
| C03 | 操作系统 | 2 |
| C04 | JAVA 程序设计 | 3 |
| C05 | 计算机组成原理 | 2 |
+----+-----+-----+
6 rows in set (0.00 sec)

mysql> delete from
    -> course
    -> where cno='C01';
ERROR 1142 (42000): DELETE command denied to user 'wrj'@'localhost' for table 'course'

mysql>
```

因为新用户只有读权限，所以可以查询数据库，但无法对数据库中的信息进行修改。

## 2.3 再次用新用户访问学生数据库，并对其中的学生表数据进行修改

登录 root 用户，授予 wrj 用户写权限：

```
MySQLExe - mysql -u root -p
C:\Windows\System32\mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 19
Server version: 5.7.21 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

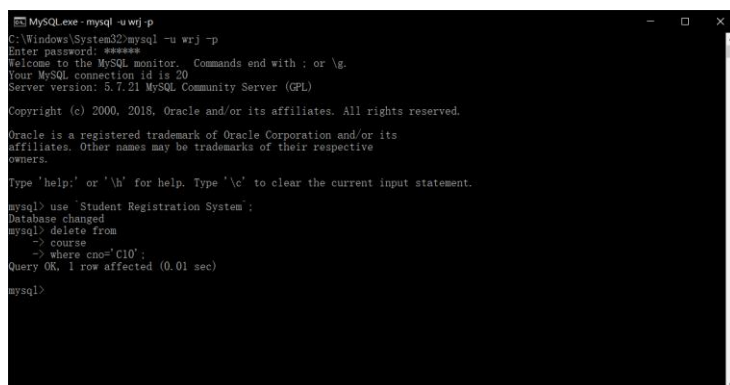
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use 'Student Registration System';
Database changed
mysql> grant delete on course to wrj;
Query OK, 0 rows affected (0.03 sec)

mysql>
```

再次用新定义的用户登录，更改课程信息，操作成功：



```
MySQL.exe - mysql -u wuj -p
C:\Windows\System32\mysql -u wrj -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 20
Server version: 5.7.21 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use 'Student Registration System';
Database changed
mysql> delete from
-> course
-> where cno='C10';
Query OK, 1 row affected (0.01 sec)

mysql>
```

## 2. 实验心得

在本次实验中，遇到的主要问题有以下三点：

- ① 存储过程的定义。起初，未更改结束符。而存储过程中的 SQL 语句以“;”为结束，将其作为定义的一部分，会报错。利用 delimiter 语句将结束符改为“//”，该问题得到解决；
- ② 触发器操作的理解。单纯使用 create trigger 语句，在 student 表上定义触发器成功。但实际操作 student 表触发触发器时，报告错误信息“it is already used by statement which invoked this stored function/trigger”。经了解，MySQL 中需要复制一个新表，然后通过对旧表的操作来触发对新表的操作。创建新的学生表后，重新定义触发器，该问题得到解决；
- ③ 新用户的创建。利用 create user 创建新用户后，在 user table 中未找到该用户。经查阅资料，变更 mysql.user 表后，需要使用 flush privileges 命令更新权限表。此命令的本质作用是将当前 user 和 privilege 表中的用户信息/权限设置提取到内存里。修改 MySQL 的用户数据和权限后，希望在不重启服务的情况下直接生效，就需要执行该命令。

实践出真知，本次数据库完整性和安全性实验是对课堂和书本所学知识的补充，存在一定的难度。通过自己动手、亲力亲为编写完整性规则和安全性相关内容的指令，加深了对其实实现方法的理解和记忆，收获颇丰。