《数据库原理》实验报告

实验名称　　数据库完整性与安全性实验

班　　级　　2014211304

学　　号　　2014211218

姓　　名　　史文翰

实验五　数据库完整性与安全性实验

一、　实验目的

1. 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。
2. 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

二、　实验环境

MySQL 5.7 on win10 x64

三、　实验内容

分为完整性实验和安全性实验。

四、　实验步骤及结果

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；

先向 sel 表中添加两个外码约束如下：

```
mysql> alter table sel add constraint foreign key (stu_no) references student(stu_no);
Query OK, 142 rows affected (0.10 sec)
Records: 142  Duplicates: 0  Warnings: 0

mysql> alter table sel add constraint foreign key (course_no) references course(course_no);
Query OK, 142 rows affected (0.07 sec)
Records: 142  Duplicates: 0  Warnings: 0
```

之后，各表的情况如下：

```
mysql> show create table sel;
+-------+---------------------------------------------------------------
| Table | Create Table
                                                                       |
+-------+---------------------------------------------------------------
| sel   | CREATE TABLE `sel` (
  `stu_no` varchar(6) NOT NULL,
  `course_no` varchar(3) NOT NULL,
  `grade` int(11) DEFAULT NULL,
  PRIMARY KEY (`stu_no`,`course_no`),
  KEY `course_no` (`course_no`),
  CONSTRAINT `sel_ibfk_1` FOREIGN KEY (`stu_no`) REFERENCES `student` (`stu_no`),
  CONSTRAINT `sel_ibfk_2` FOREIGN KEY (`course_no`) REFERENCES `course` (`course_no`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-------+---------------------------------------------------------------
1 row in set (0.01 sec)
```

```
mysql> show create table student;
+---------+------------------------------
| Table   | Create Table

+---------+------------------------------
| student | CREATE TABLE `student` (
  `stu_no` varchar(6) NOT NULL,
  `stu_name` varchar(8) DEFAULT NULL,
  `sex` varchar(2) DEFAULT NULL,
  `b_date` datetime DEFAULT NULL,
  `dept` varchar(8) DEFAULT NULL,
  `class_no` varchar(4) DEFAULT NULL,
  PRIMARY KEY (`stu_no`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+---------+------------------------------
1 row in set (0.00 sec)
```

```
mysql> show create table course;
+--------+-----------------------------------------
----------------------------------------------------
----------------------------------------+
| Table  | Create Table

                                                   |
+--------+-----------------------------------------
----------------------------------------------------
----------------------------------------+
| course | CREATE TABLE `course` (
  `course_no` varchar(3) NOT NULL,
  `course_name` varchar(12) DEFAULT NULL,
  `hours` int(11) DEFAULT NULL,
  `credit` int(11) DEFAULT NULL,
  `semester` varchar(2) DEFAULT NULL,
  PRIMARY KEY (`course_no`)
) 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，验证其实体完整性约束；

```
mysql> insert into student values('31428', '', '', '1980-1-1 00:00:00', '', '');
ERROR 1062 (23000): Duplicate entry '31428' for key 'PRIMARY'
```

```
mysql> insert into course values('C01','',0,0,'');
ERROR 1062 (23000): Duplicate entry 'C01' for key 'PRIMARY'
```

可以看出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 sel values('31428','C99',100);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`stu`.`sel`, CONSTRAINT `sel_ibfk_
2` FOREIGN KEY (`course_no`) REFERENCES `course` (`course_no`))
```

违背了外码约束，注意这里的外码的名称是系统自定义的。

(4) 删除学生表中的所有数据，验证参照完整性约束；

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

无法删除这个表，这是因为这会破坏 sel 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_sel(in stu_no_input varchar(5))
    -> begin
    -> select course_no
    -> from sel
    -> where stu_no = stu_no_input;
    -> end
    -> //
Query OK, 0 rows affected (0.02 sec)
```

注意，关键字 DELIMITER 告知命令行识别//时才进行解析执行，而不是识别到第一个分号。

```
mysql> call show_sel('31427');
+-----------+
| course_no |
+-----------+
| C01       |
| C02       |
+-----------+
2 rows in set (0.01 sec)
```

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

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

```
mysql> create trigger add_date
    -> after insert on student
    -> for each row
    -> begin
    -> update student
    -> set b_date = DATE_ADD(b_date, INTERVAL 1 DAY);
    -> end
    -> $
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> insert into student values('00000','','','0000-00-00 00:00:00','','');
ERROR 1442 (HY000): Can't update table 'student' in stored function/trigger because it is already used by statement whic
h invoked this stored function/trigger.
```

得到报错信息，经查阅资料，这是由于 MySQL 禁止向同一张表进行触发，即不能通过改变一张表来触发对这同一张表的更改。

我们可以修改 trigger 来达到目的，比如 after insert on sel

```
mysql> create trigger add_date_sel
    -> after insert on sel
    -> for each row
    -> begin
    -> update student
    -> set b_date = DATE_ADD(b_date, INTERVAL 1 DAY);
    -> end
    -> $
Query OK, 0 rows affected (0.02 sec)
```

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

之后来验证触发器执行结果：

```
| 31422 | 林昕     | 女 | 1980-01-03 00:00:00 | 计算机 | 3146 |
| 31423 | 操牡丹   | 女 | 1980-01-03 00:00:00 | 计算机 | 3146 |
| 31424 | 张乐园   | 无 | 1980-01-03 00:00:00 | 计算机 | 3146 |
| 31425 | 雷爽     | 女 | 1980-01-03 00:00:00 | 计算机 | 3146 |
| 31426 | 秦灵伶   | 女 | 1980-01-03 00:00:00 | 计算机 | 3146 |
| 31427 | 黄金花   | 女 | 1980-01-03 00:00:00 | 计算机 | 3146 |
| 31428 | 张敏     | 女 | 1980-01-03 00:00:00 | 计算机 | 3146 |
```

可以看出出生日期均+1，执行正确。

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；

```
mysql> grant select on stu.* to 'pig'@'localhost' identified by 'abc123';
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

创建了用户，只授予了其读权限。用户名为 pig，密码为 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；

这里仅演示只读权限的新用户，证明其不能对数据库进行修改。

```
mysql> select * from student;
+--------+----------+------+---------------------+--------+----------+
| stu_no | stu_name | sex  | b_date              | dept   | class_no |
+--------+----------+------+---------------------+--------+----------+
| 30201  | 吴磊     | 男   | 1980-01-02 00:00:00 | 电信   | 3022     |
| 30202  | 袁青春   | 男   | 1980-01-02 00:00:00 | 电信   | 3022     |
```

select 语句可以执行

```
mysql> insert into sel values('','','');
ERROR 1142 (42000): INSERT command denied to user 'pig'@'localhost' for table 'sel'
```

由于 pig 只拥有读权限，因此操作失败。

五、 实验小结

本实验重点是数据库的完整性约束和安全性机制，这些约束和机制隐匿于平时的操作之下，为数据库的稳定和高效提供了保证。

如何理解完整性约束？可以理解为在你对数据库做一件事的时候，这件事不是任意的，也不是没有要求的，而是要满足现有的某一些条件，这些条件就被称之为“约束”。而在数据库中约束分为很多种类，如上述验证过的主码约束，以及外码约束，都是从码的角度来定义的一大类约束。诸如触发器这样的机制为各种操作也添加了约束。