

# 《数据库系统原理》实验报告

实验名称     实验五 数据库完整性与安全性

班     级     2015211307

学     号     2017526019

姓     名     刘禾子

## 实验五 数据库完整性与安全性实验

### ■ 实验目的

1. 通过对完整性规则的定义实现，熟悉了解 MySQL 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 MySQL 中安全性的内容和实现方法，加深对数据库安全性的理解

### ■ 实验平台及环境

Windows10、MySQL 5.7、MySQL Workbench 6.3CE

### ■ 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键、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；
- (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。
- (7) 用 SQL 完成以上操作。

#### 2. 安全性实验

- (1) 定义一新的登陆账号、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，并对其中的学生表数据进行修改。
- (4) 用SQL语句分别完成以上内容。

### ■ 实验步骤及结果分析

## 1. 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；

分别建立 course、student、sc 表并定义各个表的主键

```
25 • CREATE TABLE `course` (  
26   `cno` varchar(3) NOT NULL,  
27   `cname` varchar(12) DEFAULT NULL,  
28   `lhour` int(11) DEFAULT NULL,  
29   `credit` int(11) DEFAULT NULL,  
30   `semester` varchar(2) DEFAULT NULL,  
31   PRIMARY KEY (`cno`)  
32 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
18 • CREATE TABLE `student` (  
19   `sno` varchar(6) NOT NULL,  
20   `sname` varchar(6) DEFAULT NULL,  
21   `sex` varchar(2) DEFAULT NULL,  
22   `bdate` datetime DEFAULT NULL,  
23   `dept` varchar(8) DEFAULT NULL,  
24   `classno` varchar(3) DEFAULT NULL,  
25   PRIMARY KEY (`sno`)  
26 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
25 • CREATE TABLE `sc` (  
26   `sno` varchar(6) NOT NULL,  
27   `cno` varchar(3) NOT NULL,  
28   `grade` int(11) DEFAULT NULL,  
29   PRIMARY KEY (`sno`,`cno`)  
30 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Student 主键为 sno，course 表主键为 cno，sc 表主键为 sno 和 cno，实现实体完整性约束。

```
1 • use stu;  
2 • alter table sc add constraint sno_fk foreign key(sno) references student(sno);  
3 • alter table sc add constraint cno_fk foreign key(cno) references course(cno);
```

✓	5	16:21:20	use stu	0 row(s) affected	0.000 sec
✓	6	16:21:20	alter table sc add constraint sno_fk foreign key(sno) referen...	142 row(s) affected Records: 142 Duplicates: 0 Warnings: 0	0.109 sec
✓	7	16:21:20	alter table sc add constraint cno_fk foreign key(cno) referen...	142 row(s) affected Records: 142 Duplicates: 0 Warnings: 0	0.141 sec

给 sc 表添加外键 (sno,cno) 实现参照完整性约束

分别执行 show create table student/sc/course; 显示各个表的属性以及主键和外键相关信息

1	•	show create table student;
<div> <div>Result Grid</div> <div>Filter Rows: <input type="text"/></div> <div>Export: </div> <div>Wrap Cell Content: </div> </div>		
	Table	Create Table
▶	student	<pre>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(3) DEFAULT NULL,   PRIMARY KEY (`sno`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8</pre>
1	•	show create table course;
<div> <div>Result Grid</div> <div>Filter Rows: <input type="text"/></div> <div>Export: </div> <div>Wrap Cell Content: </div> </div>		
	Table	Create Table
▶	course	<pre>CREATE TABLE `course` (   `cno` varchar(3) NOT NULL,   `cname` varchar(12) DEFAULT NULL,   `lhour` int(11) DEFAULT NULL,   `credit` int(11) DEFAULT NULL,   `semester` varchar(2) DEFAULT NULL,   PRIMARY KEY (`cno`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8</pre>
1	•	show create table sc;
<div> <div>Result Grid</div> <div>Filter Rows: <input type="text"/></div> <div>Export: </div> <div>Wrap Cell Content: </div> </div>		
	Table	Create Table
▶	sc	<pre>CREATE TABLE `sc` (   `sno` varchar(6) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL,   `cno` varchar(3) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL,   `grade` int(11) DEFAULT NULL,   PRIMARY KEY (`sno`,`cno`),   KEY `cno_fk` (`cno`),   CONSTRAINT `cno_fk` FOREIGN KEY (`cno`) REFERENCES `course` (`cno`),   CONSTRAINT `sno_fk` FOREIGN KEY (`sno`) REFERENCES `student` (`sno`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8</pre>

(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

2	•	insert into course values('C01','计算机图形学','48','5','夏');
3	•	insert into student values('30201','张三','女','1996-11-26 00:00:00','计算机','301');
<div> <div>Result Grid</div> <div>Filter Rows: <input type="text"/></div> <div>Export: </div> <div>Wrap Cell Content: </div> </div>		
18	16:50:38	insert into course values('C01','计算机图形学','48','5','夏') Error Code: 1062. Duplicate entry 'C01' for key 'PRIMARY' 0.000 sec
19	16:50:46	insert into student values('30201','张三','女','1996-11-26 00:00:00','计算机','301') Error Code: 1062. Duplicate entry '30201' for key 'PRIMARY' 0.000 sec

向 course 表和 student 表插入具有相同课程号和学号的数据，执行语句后报错，由于两个表均有实体完整性约束

- (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

```
1 • insert into sc values('31401','C10','99');
```

执行结果出错，由于不存在 C10 这个课程编号的数据

Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails

- (4) 删除学生表中的所有数据，验证参照完整性约束

```
1 • drop table student;
```

Error Code: 1217. Cannot delete or update a parent row: a foreign key constraint fails

- (5) 定义存储过程，完成查询某个学生的选课情况，并执行
- 通过 delimiter 定义分隔符，由于 MySQL 默认以 “;” 作为分隔符，若不声明分隔符则编译器会将存储过程视作 SQL 语句进行处理，则存储过程会报错，因此需实现定义分隔符，编译器不会将 “;” 当作存储过程的代码，不会去执行，待语句执行完毕后再将分隔符还原

```
1 • delimiter @@
2 • create procedure sc_info()
3 • begin select * from sc
4 • where sno='31401';
5 • end @@
```

运行存储过程

```
1 • call sc_info;
```

sno	cno	grade
31401	C01	94
31401	C02	65
31401	C03	NULL

通过 show procedure status 语句查询存储过程

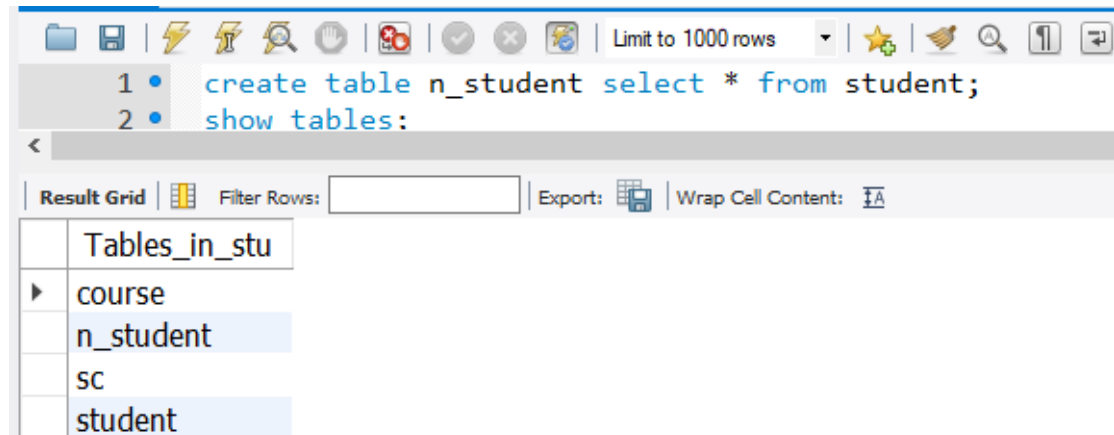
```
1 • show procedure status;
```

Db	Name	Type	Definer	Modified	Created	Security_type	Comment
stu	sc_info	PROCEDURE	root@localhost	2018-05-30 16:59:39	2018-05-30 16:59:39	DEFINER	
sys	create_synonym_db	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Takes a so
sys	diagnostics	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Create a re
sys	execute_prepared_stmt	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Takes the c
sys	ps_setup_disable_background...	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Disable all
sys	ps_setup_disable_consumer	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Disables co
sys	ps_setup_disable_instrument	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Disables in
sys	ps_setup_disable_thread	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Disable the
sys	ps_setup_enable_background_t...	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Enable all t
sys	ps_setup_enable_consumer	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Enables coi
sys	ps_setup_enable_instrument	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Enables ins
sys	ps_setup_enable_thread	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Enable the
sys	ps_setup_reload_saved	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Reloads a s
sys	ps_setup_reset_to_default	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Resets the
sys	ps_setup_save	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Saves the c
sys	ps_setup_show_disabled	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Shows all c
sys	ps_setup_show_disabled_cons...	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Shows all c
sys	ps_setup_show_disabled_instru...	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Shows all c
sys	ps_setup_show_enabled	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Shows all c
sys	ps_setup_show_enabled_consu...	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Shows all c
sys	ps_setup_show_enabled_instru...	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Shows all c
sys	ps_statement_avg_latency_hist...	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Outputs a t
sys	ps_trace_statement_digest	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Traces all i
sys	ps_trace_thread	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Dumps all
sys	ps_truncate_all_tables	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Truncates i
sys	statement_performance_analyzer	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Create a re
sys	table_exists	PROCEDURE	mysql.sys@localhost	2018-05-12 16:27:18	2018-05-12 16:27:18	INVOKER	Description Tests whet

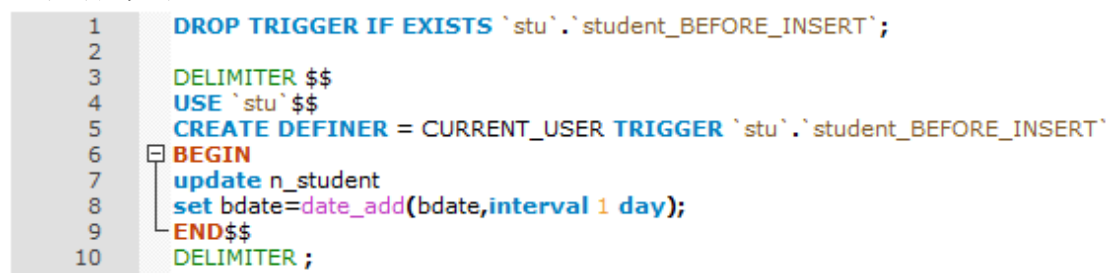
- (6) 定义触发器，当向学生表插入新的一条记录时，将所有学生出生日期加 1；并对其进行测试

由于不能通过一个表的 Insert 操作触发对同一个表进行相关操作，故这里需要重新定义一个新表 nstudent，当对 student 表进行插入后，nstudent 中的 bdate 就加 1（使用 date\_add() 函数向日期添加指定的时间间隔）。

创建新表：



定义触发器：



未执行插入语句时的搜索结果：

The screenshot shows a SQL IDE interface with a script editor containing the following SQL statement:

```
1 • SELECT * FROM stu.student;
```

The 'Result Grid' tab is active, displaying the output of the 'SELECT \* FROM stu.student;' command. The output shows a list of students in the database:

sno	sname	sex	bdate	dept	classno
30201	吴磊	男	1980-01-02 00:00:00	电信	302
30202	袁青春	男	1980-01-02 00:00:00	电信	302
30203	唐雷	男	1980-01-02 00:00:00	电信	302
30204	吴霏	男	1980-01-02 00:00:00	电信	302
30206	连洪炽	男	1980-01-02 00:00:00	电信	302
30207	王金柱	男	1980-01-02 00:00:00	电信	302
30208	苏广学	男	1980-01-02 00:00:00	电信	302
30209	唐元亮	男	1980-01-02 00:00:00	电信	302
30210	葛艳杰	男	1980-01-02 00:00:00	电信	302
30211	张永超	男	1980-01-02 00:00:00	电信	302
30212	张伟	男	1980-01-02 00:00:00	电信	302
30213	孙刚	男	1980-01-02 00:00:00	电信	302
30214	车平跃	男	1980-01-02 00:00:00	电信	302
30215	张鑫	男	1980-01-02 00:00:00	电信	302
30216	王凯	男	1980-01-02 00:00:00	电信	302
30217	马涛	男	1980-01-02 00:00:00	电信	302
30218	王佳楠	男	1980-01-02 00:00:00	电信	302
30219	马浩	男	1980-01-02 00:00:00	电信	302
30220	邵苏杰	男	1980-01-02 00:00:00	电信	302
30221	郑建华	男	1980-01-02 00:00:00	电信	302
30222	郑侃彦	男	1980-01-02 00:00:00	电信	302

插入数据后查看 n\_student 表的内容 bdate 一系列的值均加 1

1 • `insert into stu.student values('34343','张三','男','1980-01-02 00:00:00','计算机','307');`  
 2 • `SELECT * FROM n_student;`

sno	sname	sex	bdate	dept	classno
30201	吴磊	男	1980-01-03 00:00:00	电信	302
30202	袁青春	男	1980-01-03 00:00:00	电信	302
30203	唐雷	男	1980-01-03 00:00:00	电信	302
30204	吴霏	男	1980-01-03 00:00:00	电信	302
30206	连洪炽	男	1980-01-03 00:00:00	电信	302
30207	王金柱	男	1980-01-03 00:00:00	电信	302
30208	苏广学	男	1980-01-03 00:00:00	电信	302
30209	唐元亮	男	1980-01-03 00:00:00	电信	302
30210	葛艳杰	男	1980-01-03 00:00:00	电信	302

而原有的 student 表中的日期并没有+1,且插入数据成功

2  
 3 • `SELECT * FROM student;`

sno	sname	sex	bdate	dept	classno
31420	唐飞	男	1980-01-02 00:00:00	计算机	314
31422	林昕	女	1980-01-02 00:00:00	计算机	314
31423	操牡丹	女	1980-01-02 00:00:00	计算机	314
31424	张乐园	无	1980-01-02 00:00:00	计算机	314
31425	雷爽	女	1980-01-02 00:00:00	计算机	314
31426	秦灵伶	女	1980-01-02 00:00:00	计算机	314
31427	黄金花	女	1980-01-02 00:00:00	计算机	314
31428	张敏	女	1980-01-02 00:00:00	计算机	314
34343	张三	男	1980-01-02 00:00:00	计算机	307
NULL	NULL	NULL	NULL	NULL	NULL

## 2. 安全性实验

- (1) 定义一新的登陆账号、数据库用户，并授予其访问学生数据库的读权限

执行 grant 语句定义新用户名为 tackic，密码为 password

1 • `create user 'tackic'@'localhost' identified by 'password';`  
 2 • `grant select on stu.* to 'tackic'@'localhost';`  
 3 • `select host,user,authentication_string from mysql.user;`  
 4 • `flush privileges;`

host	user	authentication_string
localhost	mysql.infoschema	*THISISNOTAVALIDPASSWOR...
localhost	mysql.session	*THISISNOTAVALIDPASSWOR...
localhost	mysql.sys	*THISISNOTAVALIDPASSWOR...
localhost	root	*2470C0C06DEE42FD1618BB9...
localhost	tackic	\$A\$005\$□+ce8@□.8u~Q.R...



- (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 用户登陆：

```
C:\WINDOWS\system32\cmd.exe - mysql -uroot -p
Microsoft Windows [版本 10.0.17134.48]
(c) 2018 Microsoft Corporation。保留所有权利。

C:\Users\78914>mysql -uroot -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 22
Server version: 8.0.11 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> _
```

root 用户对 stu 数据库进行查看：

```
mysql> select * from student;
```

sno	sname	sex	bdate	dept	classno
30000	李四	男	1996-11-26 00:00:00	电信	307
30201	吴磊	男	1980-01-02 00:00:00	电信	302
30202	袁青春	男	1980-01-02 00:00:00	电信	302
30203	唐雷	男	1980-01-02 00:00:00	电信	302
30204	吴霏	男	1980-01-02 00:00:00	电信	302
30206	连洪炽	男	1980-01-02 00:00:00	电信	302
30207	王金柱	男	1980-01-02 00:00:00	电信	302
30208	苏广学	男	1980-01-02 00:00:00	电信	302
30209	唐元亮	男	1980-01-02 00:00:00	电信	302
30210	葛艳杰	男	1980-01-02 00:00:00	电信	302
30211	张永超	男	1980-01-02 00:00:00	电信	302
30212	张伟	男	1980-01-02 00:00:00	电信	302
30213	孙刚	男	1980-01-02 00:00:00	电信	302
30214	车平跃	男	1980-01-02 00:00:00	电信	302
30215	张鑫	男	1980-01-02 00:00:00	电信	302
30216	王凯	男	1980-01-02 00:00:00	电信	302
30217	马涛	男	1980-01-02 00:00:00	电信	302
30218	王佳楠	男	1980-01-02 00:00:00	电信	302
30219	马浩	男	1980-01-02 00:00:00	电信	302

root 用户对 stu 数据库进行修改：

```
mysql> update student set classno='301' where sno='30000';
Query OK, 1 row affected (0.09 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from student;
```

sno	sname	sex	bdate	dept	classno
30000	李四	男	1996-11-26 00:00:00	电信	301
30201	吴磊	男	1980-01-02 00:00:00	电信	302
30202	袁青春	男	1980-01-02 00:00:00	电信	302
30203	唐雷	男	1980-01-02 00:00:00	电信	302
30204	吴霏	男	1980-01-02 00:00:00	电信	302
30206	连洪炽	男	1980-01-02 00:00:00	电信	302
30207	王金柱	男	1980-01-02 00:00:00	电信	302
30208	苏广学	男	1980-01-02 00:00:00	电信	302
30209	唐元亮	男	1980-01-02 00:00:00	电信	302
30210	葛艳杰	男	1980-01-02 00:00:00	电信	302
30211	张永超	男	1980-01-02 00:00:00	电信	302
30212	张伟	男	1980-01-02 00:00:00	电信	302
30213	孙刚	男	1980-01-02 00:00:00	电信	302
30214	车平跃	男	1980-01-02 00:00:00	电信	302
30215	张鑫	男	1980-01-02 00:00:00	电信	302
30216	王凯	男	1980-01-02 00:00:00	电信	302
30217	马涛	男	1980-01-02 00:00:00	电信	302
30218	王佳楠	男	1980-01-02 00:00:00	电信	302
30219	马浩	男	1980-01-02 00:00:00	电信	302

登陆 tackic 用户：



```
C:\Users\78914>mysql -utackic -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 23
Server version: 8.0.11 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> _
```

tackic 对数据库进行查看:

```
mysql> use stu;
Database changed
mysql> select * from student;
```

sno	sname	sex	bdate	dept	classno
30000	李四	男	1996-11-26 00:00:00	电信	301
30201	吴磊	男	1980-01-02 00:00:00	电信	302
30202	袁青春	男	1980-01-02 00:00:00	电信	302
30203	唐雷	男	1980-01-02 00:00:00	电信	302
30204	吴霍	男	1980-01-02 00:00:00	电信	302
30206	连洪炽	男	1980-01-02 00:00:00	电信	302
30207	王金柱	男	1980-01-02 00:00:00	电信	302
30208	苏广学	男	1980-01-02 00:00:00	电信	302
30209	唐元亮	男	1980-01-02 00:00:00	电信	302
30210	葛艳杰	男	1980-01-02 00:00:00	电信	302
30211	张永超	男	1980-01-02 00:00:00	电信	302
30212	张伟	男	1980-01-02 00:00:00	电信	302
30213	孙刚	男	1980-01-02 00:00:00	电信	302

tackic 对数据库进行修改:

```
mysql> update student set classno='307' where sno='30000';
ERROR 1142 (42000): UPDATE command denied to user 'tackic'@'localhost' for table 'student'
mysql>
```

提示报错, 表明 tackic 并没有 update 权限无法修改。

- (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

登陆 root 用户对 tackic 用户授予修改权限

```
mysql> grant drop,update on stu.* to 'tackic'@'localhost';
Query OK, 0 rows affected (0.14 sec)

mysql> _
```

登陆 tackic 用户, 对数据库进行修改, 成功执行:

```
C:\Users\78914>mysql -utackic -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 25
Server version: 8.0.11 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 stu;
Database changed
mysql> update student set classno='307' where sno='30000';
Query OK, 1 row affected (0.08 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from student;
```

sno	sname	sex	bdate	dept	classno
30000	李四	男	1996-11-26 00:00:00	电信	307

## ■ 实验总结

本次实验遇到了许多问题，在添加外键时报错，后经检查发现是参照表与被参照表字符集不统一的问题，均设置成 utf8 后方可解决，在授权一块语句总是出现 42000 错误，查阅资料后将授予权限语句之后的 identified by 语句删除才可执行。此次实验我对实体完整性和参照完整性又有了更深的认识，在定义存储过程和触发器实验中，也要注意 MySQL 的许多特点，稍不注意就会出问题，在试错的过程中也了解了一些 MySQL 的特性。