

《数据库系统》 实验报告

姓名: 刘韬
学院: 竺可桢学院
专业: 人工智能
邮箱: 3220103422@zju.edu.cn

报告日期: 2024 年 3 月 29 日

实验步骤

1. 定义若干表，其中包括 **primary key**, **foreign key** 和 **check** 的定义。

```
use lab3;

create table student(
  sid char(9) primary key,
  sname char(8),
  ssex char(2),
  sage int,
  major char(4),
  check(ssex in ('M', 'F')),
  check(sage >= 10 and sage <= 100)
);

create table enroll(
  sid char(9),
  cid char(8),
  grade int,
  primary key(sid, cid),
  foreign key(sid) references student(sid)
);
```

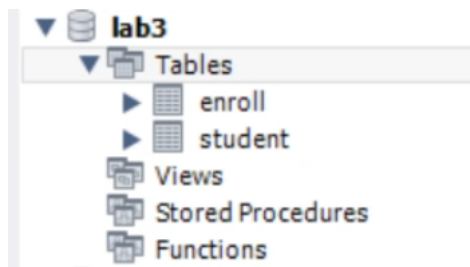


图 1: 创建表结果

2. 让表中插入数据，考察 **primary key** 如何控制实体完整性。

```
insert student values('1001', 'Wang', 'M', 20, 'CS');
insert student values('1002', 'Li', 'F', 19, 'CS');
insert student values('1003', 'Zhang', 'M', 21, 'EE');
insert student values('1004', 'Zhao', 'F', 22, 'CS');
```

再插入一条主键 id 重复的记录

```
insert student values('1001', 'Ping', 'F', 22, 'CE');
```

会发生

Error Code: 1062. Duplicate entry '1001' for key 'student.PRIMARY'

| | | | |
|---|----------|--|--|
| 7 | 14:07:59 | insert student values('1004', 'Zhao', 'F', 22, 'CS') | 1 row(s) affected |
| 8 | 14:08:07 | insert student values('1001', 'Ping', 'F', 22, 'CE') | Error Code: 1062. Duplicate entry '1001' for key 'student.PRIMARY' |

图 2: 插入重复主键的结果

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。

在 enroll 表中插入一条引用表 students 中主键为"1001"的记录

```
insert enroll values('1001', 'C1', 90);
```

随后删除 students 表中名为 wang 的记录，其 sid 为"1001"

```
delete from student where sname = 'Wang';
```

报错: Error Code: 1175. You are using safe update mode and you tried to update a table without a WHERE that uses a KEY column. To disable safe mode, toggle the option in Preferences -> SQL Editor and reconnect. 0.000 sec

这是由于 mysql safe mode 导致的。关闭 safe mode 后得到以下报错

| | | | |
|---|----------|--|---|
| 4 | 14:29:20 | delete from student where sname = 'Wang' | Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (lab3.enroll, CONSTRAINT enroll_ibfk_1 FOREIGN KEY (sid) REFERENCES student (sid)) |
|---|----------|--|---|

图 3: 删除被引用表中的行的结果

Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (lab3.enroll, CONSTRAINT enroll_ibfk_1 FOREIGN KEY (sid) REFERENCES student (sid))

这体现了 foreign key 中 on delete 子句如何控制参照完整性: 只有先将表 enroll 中引用"1001"的记录删除后，才能删除 student 表中主键为"1001"的记录。

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

```
update student set sid = '1005' where sname = 'wang';
```

| | | | |
|---|----------|--|---|
| 5 | 14:29:48 | update student set sid = '1005' where sname = 'wang' | Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (lab3.enroll, CONSTRAINT enroll_ibfk_1 FOREIGN KEY (sid) REFERENCES student (sid)) |
|---|----------|--|---|

图 4: 修改被引用表中的行的 primary key 的结果

Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (lab3.enroll, CONSTRAINT enroll_ibfk_1 FOREIGN KEY (sid) REFERENCES student (sid))

5. 修改或插入表中数据，考察 check 子句如何控制校验完整性。

```
insert student values('1005', 'Wang', 'M', -20, 'CS')
```

| | | | |
|---|----------|---|---|
| 6 | 14:34:25 | insert student values('1005', 'Wang', 'M', -20, 'CS') | Error Code: 3819. Check constraint 'student_chk_2' is violated. |
|---|----------|---|---|

图 5: 插入不符合 check 约束的数据的结果

报错: Error Code: 3819. Check constraint 'student_chk_2' is violated. 0.015 sec

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。

```
create assertion check_age check (not exists (select * from student where sage >= 50 and sage <= 100));
```

但 mysql 不支持 assertion, 所以。。。



```
create assertion check_age check (not exists (select * from students  
where age>50));
```

图 6: 定义断言的结果

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。

| | sid | sname | ssex | sage | major |
|---|------|-------|------|------|-------|
| ▶ | 1001 | Wang | M | 20 | CS |
| | 1002 | Li | F | 19 | CS |
| | 1003 | Zhang | M | 21 | EE |
| | 1004 | Zhao | F | 22 | CS |
| | 1005 | Wang | M | 20 | CS |
| ★ | NULL | NULL | NULL | NULL | NULL |

图 7: 原先的 student

| | sid | cid | grade |
|---|------|-----|-------|
| ▶ | 1001 | C1 | 90 |

图 8: 原先的 enroll

定义下面的 trigger

```
Delimiter $$  
Create trigger age_present  
After update on student  
For each row  
Begin  
  Update enroll set grade =100  
  where enroll.sid in (select sid from student where sage<20);  
end; $$  
Delimiter ;
```

```
mysql> select * from enroll;
+-----+-----+-----+
| sid | cid | grade |
+-----+-----+-----+
| 1001 | C1 | 100 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from student
-> ;
+-----+-----+-----+-----+-----+
| sid | sname | ssex | sage | major |
+-----+-----+-----+-----+-----+
| 1001 | Wang | M | 12 | CS |
| 1002 | Li | F | 12 | CS |
| 1003 | Zhang | M | 21 | EE |
| 1004 | Zhao | F | 22 | CS |
| 1005 | Wang | M | 12 | CS |
| 1007 | Zhao | F | 22 | CS |
| 1008 | Sun | M | 20 | CS |
+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

图 9: 定义触发器的结果

可以看到 trigger 起作用了, 当 student 表中的 sage 小于 20 时, enroll 表中的 grade 被更新为 100。

8. 遇到的问题和解决办法

在开始写 trigger 定义时使用的是

```
Update enroll set grade =100
where enroll.sid in (select sid from student where sage<20);
```

在定义 trigger 后, update student table 出现 error 1146 : Table 'lab3.students' doesn't exist.

```
mysql> select * from student;
+-----+-----+-----+-----+-----+
| sid | sname | ssex | sage | major |
+-----+-----+-----+-----+-----+
| 1001 | Wang | M | 20 | CS |
| 1002 | Li | F | 19 | CS |
| 1003 | Zhang | M | 21 | EE |
| 1004 | Zhao | F | 22 | CS |
| 1005 | Wang | M | 20 | CS |
| 1007 | Zhao | F | 22 | CS |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> insert student value ('1008','Sun','M','20','CS');
Query OK, 1 row affected (0.01 sec)

mysql> update student set sname = "kun" where sid = '1002';
ERROR 1146 (42S02): Table 'lab3.students' doesn't exist
```

图 10: 遇到的问题

这是因为这里触发了 trigger, 而且 trigger 内的 sql 语句中的表名写错了, 应该是 student 而不是 students。所以报错是因为找不到 students 表。解决方案就是把 trigger 修正。

9. 总结

lab3 验证了 primary key, foreign key, check, assertion, trigger 的作用, 以及如何通过这些约束来保证数据的完整性。同时也遇到了一些问题, 比如 mysql 不支持 assertion, 以及在定义 trigger 时表名写错的问题。这些告诉我使用数据库时要注意严谨, 不能出现错误偏差, 否则会产生意外的 error。