

《数据库概论》 实验二 用户自定义完整性约束 实验报告

姓名 王明 学号 161220124 联系方式 1090616897@qq.com

实验环境

我使用的操作系统是 windows10、使用的数据库软件是 MySQL WorkBench8.0.12。

实验过程

0、创建数据库 exp2:

```
1 • create database exp2;  
2 • use exp2;
```

1、使用 SQL 语句建立基本表(Stuff, Section, Project, Job)并插入数据:

```
6 • create table Stuff(  
7     StuffName char(8),  
8     StuffNo int,  
9     Age int,  
10    Salary int,  
11    SectionNo int);  
12 • create table Section(  
13     SectionName char(20),  
14     SectionNo int,  
15     StuffInCharge int);  
16 • create table Project(  
17     ProjectName char(20),  
18     ProjectNo int,  
19     SectionInCharge int);  
20 • create table Job(  
21     StuffNo int,  
22     ProjectNo int,  
23     WorkTime int);  
24 • insert into Stuff values('Lily',1,27,30000,1);  
25 • insert into Stuff values('Alice',2,26,32000,2);  
26 • insert into Stuff values('Tom',3,29,27000,4);  
27 • insert into Stuff values('Jim',4,25,31000,4);  
28 • insert into Stuff values('Jack',5,33,46000,2);  
29 • insert into Stuff values('Molly',6,30,40000,3);  
30 • insert into Section values('Personnel',1,1);  
31 • insert into Section values('Sales',2,5);  
32 • insert into Section values('Advertising',3,6);  
33 • insert into Section values('Research',4,3);  
34 • insert into Project values('StuffReduce',1,1);  
35 • insert into Project values('PricePromotion',2,2);  
36 • insert into Project values('Branding',3,3);  
37 • insert into Project values('Excelsior',4,4);  
38 • insert into Job values(1,1,23);  
39 • insert into Job values(2,2,13);  
40 • insert into Job values(3,4,29);  
41 • insert into Job values(4,4,24);  
42 • insert into Job values(5,2,18);  
43 • insert into Job values(6,3,30);
```

结果:

	StuffName	StuffNo	Age	Salary	SectionNo
▶	Lily	1	27	30000	1
	Alice	2	26	32000	2
	Tom	3	29	27000	4
	Jim	4	25	31000	4
	Jack	5	33	46000	2
	Molly	6	30	40000	3

	SectionName	SectionNo	StuffInCharge
▶	Personnel	1	1
	Sales	2	5
	Advertising	3	6
	Research	4	3

	ProjectName	ProjectNo	SectionInCharge
▶	StuffReduce	1	1
	PricePromotion	2	2
	Branding	3	3
	Excelsior	4	4

	StuffNo	ProjectNo	WorkTime
▶	1	1	23
	2	2	13
	3	4	29
	4	4	24
	5	2	18
	6	3	30

2、使用 SQL 语句为基本表添加约束:

◆ 2.1. 添加主键约束。

```
50 • alter table Stuff add primary key(StuffNo);
51 • alter table Section add primary key(SectionNo);
52 • alter table Project add primary key(ProjectNo);
53 • alter table Job add primary key(StuffNo,ProjectNo);
```

◆ 2.2. 添加非空约束。

```
54 • alter table Stuff modify StuffNo int not null;
55 • alter table Section modify SectionNo int not null;
56 • alter table Project modify ProjectNo int not null;
57 • alter table Job modify StuffNo int not null;
58 • alter table Job modify ProjectNo int not null;
```

◆ 2.3. 添加外键约束。

```
59 • alter table Stuff add constraint FK1 foreign key FK1(SectionNo)
60   references Section(SectionNo) on update cascade on delete restrict;
61 • alter table Section add constraint FK2 foreign key FK2(StuffInCharge)
62   references Stuff(StuffNo) on update cascade on delete restrict;
63 • alter table Project add constraint FK3 foreign key FK3(SectionInCharge)
64   references Section(SectionNo) on update cascade on delete restrict;
65 • alter table Job add constraint FK4 foreign key FK4(StuffNo)
66   references Stuff(StuffNo) on update cascade on delete restrict;
67 • alter table Job add constraint FK5 foreign key FK5(ProjectNo)
68   references Project(ProjectNo) on update cascade on delete restrict;
```

3、设计违反数据约束的操作：

◆ 3.1. 违反非空约束。

```
71 • insert into Stuff values('Bob',null,20,20000,1);
```

结果：

```
✖ 123 20:56:07 insert into Stuff values('Bob',null,20,20000,1)
```

Error Code: 1048. Column 'StuffNo' cannot be null

◆ 3.2. 违反外键约束。

```
72 • insert into Stuff values('Bob',7,20,20000,5);
```

结果：

```
✖ 124 20:59:11 insert into Stuff values('Bob',7,20,20000,5)
```

Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('exp2'.stuff, CONSTRAINT 'FKC1' FOREIGN KEY ('SectionNo' REFERENCES 'exp2'.section))

```
73 • update Section set StuffInCharge = 7 where SectionNo = 1;
```

结果：

```
✖ 125 21:02:55 update Section set StuffInCharge = 7 where SectionNo = 1
```

Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('exp2'.section, CONSTRAINT 'FKC2' FOREIGN KEY ('StuffInCharge' REFERENCES 'exp2'.stuff))

```
74 • delete from Section where SectionNo = 1;
```

结果：

```
✖ 126 21:06:28 delete from Section where SectionNo = 1
```

Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails ('exp2'.stuff, CONSTRAINT 'FKC1' FOREIGN KEY ('SectionNo' REFERENCES 'exp2'.section))

```
75 • delete from Stuff where Stuffno = 1;
```

结果：

```
✖ 127 21:12:15 delete from Stuff where Stuffno = 1
```

Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails ('exp2'.section, CONSTRAINT 'FKC2' FOREIGN KEY ('StuffInCharge' REFERENCES 'exp2'.stuff))

4、创建触发器根据新加入的项目对工作时间和年薪进行修改：

◆ 4.1. 触发器创建。

```
78 delimiter $$
79 • create trigger trigger1 before insert on Job
80 for each row
81 begin
82     if new.WorkTime > 24 then set new.WorkTime = 24;
83     end if;
84     update Stuff set Stuff.Salary = Stuff.Salary * (1 + 0.05)
85     where Stuff.StuffNo = new.StuffNo;
86     update Stuff set Stuff.Salary = Stuff.Salary * (1 + 0.03)
87     where Stuff.StuffNo = new.StuffNo
88     and new.StuffNo in (select StuffInCharge from Section);
89 end$$
90 delimiter ;
```

◆ 4.2. 设计数据来检验触发器的功能。

```
93 • insert into Job values(6,2,60);
```

```
94 • insert into Job values(2,1,44);
```

结果：

插入前的 Job 表和 Stuff 表：

	StuffNo	ProjectNo	WorkTime
▶	1	1	23
	2	2	13
	3	4	29
	4	4	24
	5	2	18
	6	3	30

	StuffName	StuffNo	Age	Salary	SectionNo
▶	Lily	1	27	30000	1
	Alice	2	26	32000	2
	Tom	3	29	27000	4
	Jim	4	25	31000	4
	Jack	5	33	46000	2
	Molly	6	30	40000	3

插入后的 Job 表和 Stuff 表：

	StuffNo	ProjectNo	WorkTime
▶	1	1	23
	2	1	24
	2	2	13
	3	4	29
	4	4	24
	5	2	18
	6	2	24
	6	3	30
★	NULL	NULL	NULL

	StuffName	StuffNo	Age	Salary	SectionNo
▶	Lily	1	27	30000	1
	Alice	2	26	33600	2
	Tom	3	29	27000	4
	Jim	4	25	31000	4
	Jack	5	33	46000	2
	Molly	6	30	43260	3
★	NULL	NULL	NULL	NULL	NULL

可以看出 Job 表中多了(2,1,24)(6,2,24)实现了工作时间多于 24 就设置为 24 的功能。可以看出 Stuff 表中 2 号员工的年薪增长了 5%，而 6 号员工同时是一个部门的负责人，所以 6 号的年薪先增长 5%，再增长 3%。所以，经检验，触发器的功能正确。

5、创建新用户 worker 并授权：

◆ 5.1.创建新用户并授予一定的权限（最后删除该用户）。

```

106 ● create user 'worker'@'localhost' identified by 'wmfrank';
107 ● grant select, update(Age) on Stuff to 'worker'@'localhost';
108 ● drop user 'worker'@'localhost';

```

◆ 5.2.登录新用户来对数据库进行修改。


```

1 • use exp2;
2 • update Stuff set Age = 22 where StuffNo = 1;
3 • select * from Stuff;
4 • update Stuff set Salary = 60000 where StuffNo = 5;
5 • select * from Stuff;

```

结果:

修改前的 Stuff 表:

	StuffName	StuffNo	Age	Salary	SectionNo
▶	Lily	1	27	30000	1
	Alice	2	26	33600	2
	Tom	3	29	27000	4
	Jim	4	25	31000	4
	Jack	5	33	46000	2
	Molly	6	30	43260	3
*	NULL	NULL	NULL	NULL	NULL

修改后的 Stuff 表:

	StuffName	StuffNo	Age	Salary	SectionNo
▶	Lily	1	22	30000	1
	Alice	2	26	33600	2
	Tom	3	29	27000	4
	Jim	4	25	31000	4
	Jack	5	33	46000	2
	Molly	6	30	43260	3
*	NULL	NULL	NULL	NULL	NULL

修改 1 号员工的年龄成功。

✖ 5 00:40:03 update Stuff set Salary = 60000 where StuffNo = 5
 Error Code: 1143. UPDATE command denied to user 'worker'@'localhost' for column 'Salary' in table 'stuff'

修改 6 号员工的年薪失败，因为只有年龄列的修改权，没有年薪列的修改权。

实验中遇到的困难及解决办法

Q1:设置触发器的时候总是遇到语法错误，无论怎么改都不对，一度感觉崩溃。
 S1:后来在网上学习了 delimiter 的用法，原来 MySQL 是根据终止符来判断语句的终止的，默认的终止符是分号;，而触发器的语块中包含了其他的语句，因此编译器无法分辨语句的终止，这时候就需要 delimiter 来重新定义终止符（我定义的是 \$\$）方便编译器辨认，语句执行完后再用 delimiter 修改终止符为分号。如下图所示：

```

78 delimiter $$
79 • create trigger trigger1 before insert on Job
80   for each row
81   begin
82     if new.WorkTime > 24 then set new.WorkTime = 24;
83   end if;
84   update Stuff set Stuff.Salary = Stuff.Salary * (1 + 0.05)
85   where Stuff.StuffNo = new.StuffNo;
86   update Stuff set Stuff.Salary = Stuff.Salary * (1 + 0.03)
87   where Stuff.StuffNo = new.StuffNo
88     and new.StuffNo in (select StuffInCharge from Section);
89 end$$
90 delimiter ;

```

参考文献及致谢

参考网站：

<https://blog.csdn.net/a909301740/article/details/62887992>

<https://www.cnblogs.com/phpper/p/7587031.html>

致谢：非常感谢两位博主的分享，让我对于语句的约束以及触发器的内容有了深入的了解，对实验的完成非常有帮助。