

# 《数据库系统原理》实验报告

( 实验名称 : SQL 完整性 )

专    业	电气工程及其自动化
班    级	电气 1904
学    号	3190102182
学生姓名	何宇昊
指导老师	林怀忠

浙江大学

2021 年 5 月 11 日

## 一、实验目的

本次实验的实验目的主要在于考查主键，外键如何约束完整性，以及考查触发器在数据库中的作用。

## 二、实验内容：

1. 了解参照完整性中外键设置的 on delete cascade/set null/no action (默认, 同 restrict) 的原理和区别, on update 也是类似的。
2. 通过实验4提供的sql脚本创建表和插入数据 (自己可以看看lab4.sql脚本中的sql语句)
  - a) 考查主键完整性约束, 在employee表中插入一名名为Betty (主码) 的员工 (通常我们不会把名字作为主键, 考虑重名, 一般是在前面加一个id字段, 这里不作为考虑)
  - b) 考查 on delete no action/ restrict, 删除employee表中员工名字为Betty的数据, 发现error (因为带有外键约束不能删除)
  - c) 接下来我们需要创建 (或者修改) 另外的几个外键 (可以通过直接在sql脚本修改以及新添加外键约束的sql语句, 然后直接重新运行脚本; 也可以在数据库中删除原有的外键, 然后把新的外键约束通过alter命令添加)
    - i. works表中的employee\_name 参照 父表employee中的employee\_name 在删除时级联删除, 更新时级联更新,
    - ii. works表中的company\_name参照父表company中的company\_name在删除时设置为null, 更新时级联更新,
    - iii. manages表中的employee\_name参照父表employee中的

employee\_name 在删除时级联删除，更新时级联更新，

iv. manages表中的manager\_name参照父表employee中的

employee\_name 在删除时设置为null，更新时级联更新，

d) 接下来再删除employee表中员工名字为Betty的数据，查看另外几个表关于Betty的数据有什么变化。

e) 修改员工表中姓名为Amanda的员工名字为Amy，查看另外几个表原本关于Amanda的数据有什么变化。

f) 考查触发器，这里我们在company表中增加一条属性tot\_salary，也就是本公司所以在职的员工总工资，通过写一条更新sql语句更新tot\_salary值，最后创建三个触发器，分别是在insert，update，delete works表数据后，对相应的tot\_salary进行更新操作。（exp：当插入一条A公司的员工数据时候，此员工薪水为2000，则A公司总薪水需要加上2000。）

### 三、实验环境

系统环境：Ubuntu 20.04 LTS Focal;

Mysql 版本：Ver 8.0.23-0ubuntu0.20.04.1 for Linux on x86\_64 ((Ubuntu))

### 四、程序源码与运行结果

实验任务 2：

a. 尝试加入（失败）

```
1 mysql haoge@(none):db2021_lab4> insert into employee(employee_name) values("Betty");
2 (1062, "Duplicate entry 'Betty' for key 'employee.PRIMARY'")
```

b. 尝试删除（失败）

```

1 mysql haoge@none:db2021_lab4> delete from employee where employee_name = "Betty";
2 You're about to run a destructive command.
3 Do you want to proceed? (y/n): y
4 Your call!
5 (145), Cannot delete or update a parent row: a foreign key constraint fails ('db2021_lab4','works', CONSTRAINT `works_ibfk_1` FOREIGN KEY (`employee_name`) REFERENCES `employee` (`employee_name`))

```

c. 调整外键设置（在 lab4.sql 文件中调整）

```

create table works(
    employee_name varchar(20),
    company_name varchar(20),
    salary int,
    primary key(employee_name),
    -- 这里我只定义了一个外键, 而且是no action, 等同于 restrict, 当在外键的来源表中删除
    foreign key (employee_name) references employee(employee_name)
        on delete cascade
        on update cascade,
    foreign key (company_name) references company(company_name)
        on delete set null
        on update cascade
)ENGINE=InnoDB DEFAULT CHARSET=utf8;

create table manages(
    employee_name varchar(20),
    manager_name varchar(20),
    primary key(employee_name),
    foreign key ( employee_name ) references employee(employee_name)
        on delete cascade
        on update cascade,
    foreign key(manager_name) references employee(employee_name)
        on delete set null
        on update cascade
)ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

d. 各表输出

employee:

employee_name	street	city
Amanda	King	Birmingham
Dean	Dover	Sheffield
Kevin	King	Birmingham
Lily	Regent	Nottingham
Mark	Dover	Sheffield
Richard	Strand	Cambridge
Whitney	Regent	Nottingham

company:

company_name	city
Bertelsmann	Cambridge
Chloris	Nottingham
Eldat GmbH	Birmingham
Greenteam	Cambridge

works:

employee_name	company_name	salary
Amanda	Greenteam	2800
Dean	Eldat GmbH	3100
Kevin	Greenteam	2700
Lily	Eldat GmbH	3800
Mark	Chloris	4000
Richard	Bertelsmann	3000
Whitney	Chloris	2500

manages:

employee_name	manager_name
Richard	<null>
Kevin	Amanda
Dean	Lily
Whitney	Mark

e. 各表输出

employee:			
employee_name	street	city	
Amy	King	Birmingham	
Dean	Dover	Sheffield	
Kevin	King	Birmingham	
Lily	Regent	Nottingham	
Mark	Dover	Sheffield	
Richard	Strand	Cambridge	
Whitney	Regent	Nottingham	

  

company:		
company_name	city	
Bertelsmann	Cambridge	
Chloris	Nottingham	
Eldat GmbH	Birmingham	
Greenteam	Cambridge	

  

works:			
employee_name	company_name	salary	
Amy	Greenteam	2800	
Dean	Eldat GmbH	3100	
Kevin	Greenteam	2700	
Lily	Eldat GmbH	3800	
Mark	Chloris	4000	
Richard	Bertelsmann	3000	
Whitney	Chloris	2500	

  

manages:		
employee_name	manager_name	
Richard	<null>	
Kevin	Amy	
Dean	Lily	
Whitney	Mark	

#### f. 表格更新+触发器设置

a) 创建了一个视图，利用视图对 company 更新

```

1 alter table company add tot_salary int;
2 create view tmp as select company_name,sum(salary) as s from works group by company_name;
3 update company set tot_salary = (select s from tmp where company.company_name = tmp.company_name);

```

b) 结果

company_name	city	tot_salary
Bertelsmann	Cambridge	3000
Chloris	Nottingham	6500
Eldat GmbH	Birmingham	6900
Greenteam	Cambridge	5500

c) 触发器设置

```
create trigger `Tri_del`
after insert on `works`
for each row
update company set tot_salary = tot_salary + new.salary where(company_name = new.company_name);

create trigger `Tri_ins`
after insert on `works`
for each row
update company set tot_salary = tot_salary + new.salary where(company_name = new.company_name);

create trigger `Tri_upd`
after update on `works`
for each row
update company set tot_salary = tot_salary + new.salary - old.salary where(company_name=new.company_name);
```

d) 触发器调试

i. 数据更新前的 company 表

company_name	city	tot_salary
Bertelsmann	Cambridge	3000
Chloris	Nottingham	6500
Eldat GmbH	Birmingham	6900
Greenteam	Cambridge	5500

ii. 进行数据操作(insert)

```
insert into employee values ('A','GZ','China'),('B','HZ','China');
insert into works() values ('A','Chloris',20),('B','Bertelsmann',10);
```

iii. company 表



company_name	city	tot_salary
Bertelsmann	Cambridge	3010
Chloris	Nottingham	6520
Eldat GmbH	Birmingham	6900
Greenteam	Cambridge	5500

iv. 进行 delete 操作 ( 值得注意的是, 如果是删除 employee 表中的信息导致的级联删除, 在 mysql8.0 版本中并不会触发 trigger )

```
delete from works where employee_name = 'B';
```

v. company 表

company_name	city	tot_salary
Bertelsmann	Cambridge	3000
Chloris	Nottingham	6520
Eldat GmbH	Birmingham	6900
Greenteam	Cambridge	5500

vi. 进行 update 操作

```
update works set salary = 31 where employee_name = 'A';
```

vii. company 表

company_name	city	tot_salary
Bertelsmann	Cambridge	3000
Chloris	Nottingham	6531
Eldat GmbH	Birmingham	6900
Greenteam	Cambridge	5500

## 五、实验总结

这次实验中有相当一部分命令是在 *mysql 8.0 Manual* 的帮助下完成的。再再再次体会到了寻找帮助文件的重要性。此外, 通过这次实验我还较为系统地学习了如何设置外键 *reference*



并且进行修改，对于 update 的使用也更加熟练。

有一点不足的地方在于，对 works 表增加属性并且更新的时候采取了一种比较笨的方法进行数据的更新。或许可以使用一行命令来执行。

在练习触发器的时候，发现一点非常重要：在 mysql8.0 中，级联删除并不会触发触发器。需要通过额外的代码（在表 A 中设置删除会触发 B 表中触发器的触发器）来实现。