

《数据库系统原理》实验报告

(实验名称：SQL 数据完整性)

专 业	<u>信息资源管理</u>
班 级	<u>信管 1601</u>
学 号	<u>3160104494</u>
学生姓名	<u>梁 鑫 宇</u>
指导老师	<u>林 怀 忠, 田 若 言</u>

浙江大学

2018 年 04 月 11 日

一、实验目的

掌握 SQL 数据完整性控制的方法。

二、实验内容：

- 定义若干表，其中包括 Primary Key, Foreign Key(on delete cascade、on update cascade)和 Check 的定义。
- 在表中插入数据，一般可以通过自己插入 3-6 条典型数据用于考察 Primary Key 如何控制实体完整性。
- 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。
- 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。
- 修改或插入表中数据，考察 Check 子句如何控制校验完整性。（MySQL 目前仅支持 Check 子句解析，但在执行过程中会忽略 Check 子句，请按实际结果完成实验报告）
- 定义一个 Assertion，并通过修改表中数据考察断言如何控制数据完整性。（MySQL/SQL Server 目前不支持 Assertion，希望能掌握概念，本实验中不要求体现）
- 定义一个 Trigger，并通过修改表中数据考察触发器如何起作用。
- 完成实验报告。要求附带实验中相关步骤的截图。

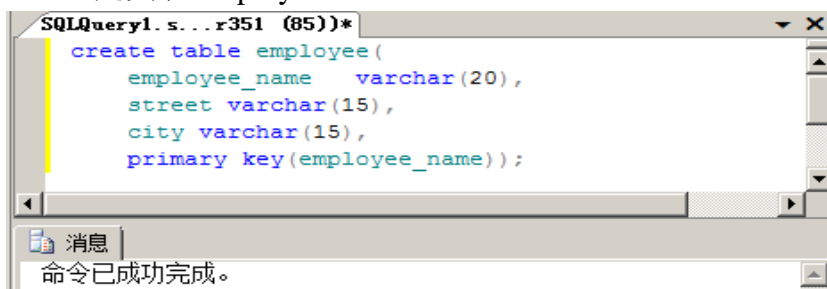
三、实验环境

数据库管理系统：SQL Server 2008

四、程序源码与运行结果

- 定义若干表，其中包括 Primary Key, Foreign Key(on delete cascade、on update cascade)和 Check 的定义。

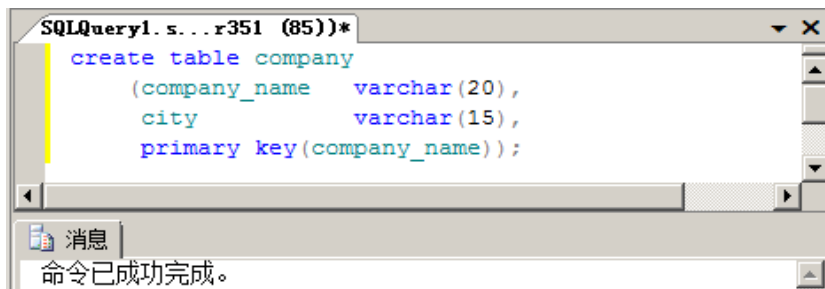
(1) 定义表 employee



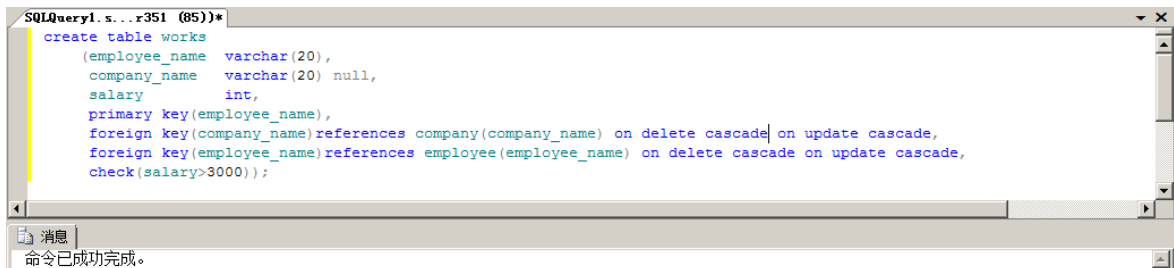
```
SQLQuery1. s...r351 (85))*
create table employee(
    employee_name varchar(20),
    street varchar(15),
    city varchar(15),
    primary key(employee_name));
```

消息
命令已成功完成。

(2) 定义表 company



(3) 定义表 works

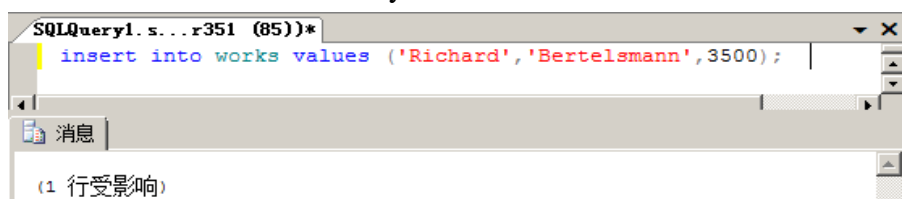


● 在表中插入数据，一般可以通过自己插入 3-6 条典型数据用于考察 Primary Key 如何控制实体完整性。

(1) 由于向 works 表中插入的第 6 条数据不满足 check 约束条件 salary>3000，于是系统报错

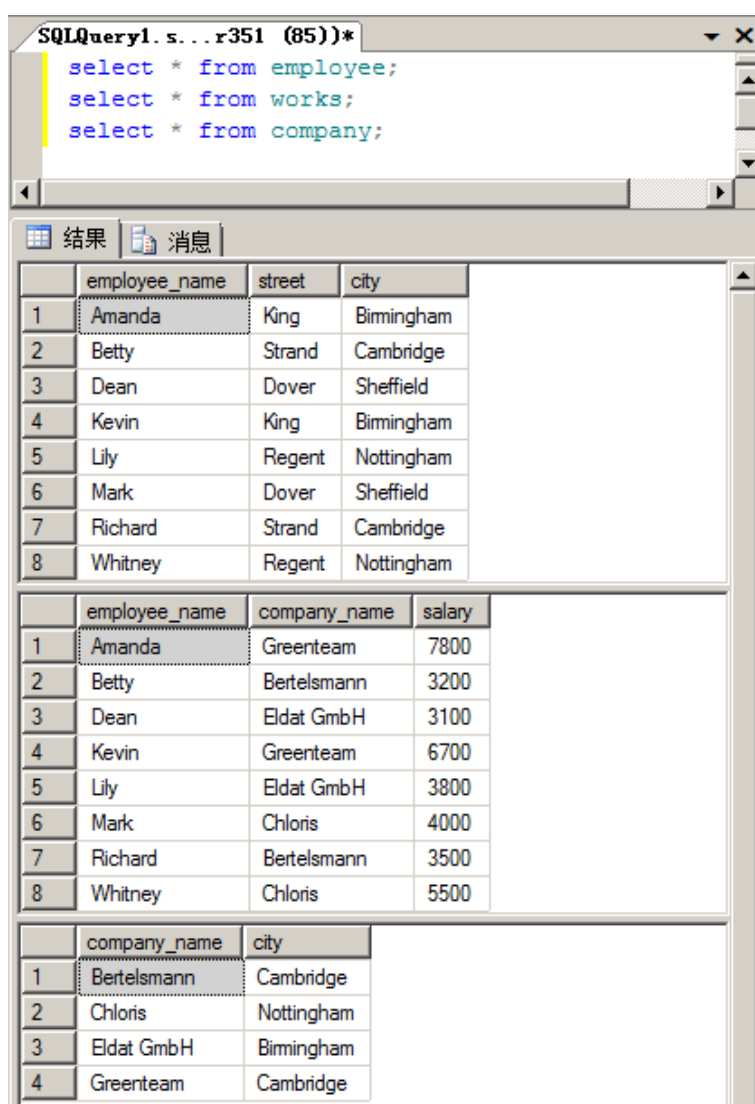


(2) 修改此条记录的 salary 值重新插入



● 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。

(1) 先展示删除前的数据



(2) 删除操作完成后

SQLQuery1.s...r351 (85))*

```

delete from employee
where employee_name = 'Richard';

select * from employee;
select * from works;

```

结果 消息

	employee_name	street	city
1	Amanda	King	Birmingham
2	Betty	Strand	Cambridge
3	Dean	Dover	Sheffield
4	Kevin	King	Birmingham
5	Lily	Regent	Nottingham
6	Mark	Dover	Sheffield
7	Whitney	Regent	Nottingham

	employee_name	company_name	salary
1	Amanda	Greenteam	7800
2	Betty	Bertelsmann	3200
3	Dean	Eldat GmbH	3100
4	Kevin	Greenteam	6700
5	Lily	Eldat GmbH	3800
6	Mark	Chloris	4000
7	Whitney	Chloris	5500

- 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。

SQLQuery1.s...r351 (85))*

```

UPDATE employee SET employee_name = 'Lucy'
WHERE employee_name = 'Lily';

select * from employee;
select * from works;

```

结果 消息

	employee_name	street	city
1	Amanda	King	Birmingham
2	Betty	Strand	Cambridge
3	Dean	Dover	Sheffield
4	Kevin	King	Birmingham
5	Lucy	Regent	Nottingham
6	Mark	Dover	Sheffield
7	Whitney	Regent	Nottingham

	employee_name	company_name	salary
1	Amanda	Greenteam	7800
2	Betty	Bertelsmann	3200
3	Dean	Eldat GmbH	3100
4	Kevin	Greenteam	6700
5	Lucy	Eldat GmbH	3800
6	Mark	Chloris	4000
7	Whitney	Chloris	5500

- 修改或插入表中数据，考察 Check 子句如何控制校验完整性。
 - (1) 插入数据 check 起作用在上文插入中已经展示。
 - (2) 修改表中数据



- 定义一个 Trigger, 并通过修改表中数据考察触发器如何起作用。

```
create trigger check_total_salary_update
on works
after update
as
begin
    update company
        set company.total_salary = (select sum(salary)
                                    from works as R
                                    group by company_name
                                    having R.company_name = company.company_name)
end
```

SQLQuery1.sql - 1...35 (user351 (56))* x SQLQuery2.sql - 1...35 (user351 (57))*

```
select * from works;
select * from company;

update works
set salary = 5000
where employee_name='Dean';

select * from works;
select * from company;
```

100 %

结果 消息

	employee_name	company_name	salary
1	Amanda	Greenteam	10000
2	Betty	Bertelsmann	3200
3	Dean	Eldat GmbH	3100

	company_name	city	total_salary
1	Bertelsmann	Cambridge	3200
2	Chloris	Nottingham	9500
3	Eldat GmbH	Birmingham	6900

	employee_name	company_name	salary
1	Amanda	Greenteam	10000
2	Betty	Bertelsmann	3200
3	Dean	Eldat GmbH	5000

	company_name	city	total_salary
1	Bertelsmann	Cambridge	3200
2	Chloris	Nottingham	9500
3	Eldat GmbH	Birmingham	8800

五、实验总结

1 . 实验中遇到的问题及解决过程

on delete 语句不支持 set null，未知原因；trigger 未掌握 deleted 表等用法，反复尝试许久才完成

2 . 实验体会和收获。

多查资料多学习，反复尝试。另外不要把一些操作想得太复杂，反而阻碍了任务的完成