# 数据库实验报告

# 1. 实验目的

熟悉数据库的数据更新操作，能够使用SQL语句对数据库进行数据的插入、更新、删除操作。

# 2.实验环境

* 数据库管理系统 PostgreSQL

* 包含所需表和数据的数据库实例。

在该数据库中存在4张表格，分别为：

STUDENTS(sid,sname,email,grade)

TEACHERS(tid,tname,email,salary)

COURSES(cid,cname,hour)

CHOICES(no,sid,tid,cid,score)

# 3.实验常用语句

用于在数据库中进行数据的插入、更新和删除操作：

### 插入（Insert）

* 插入完整的行：

INSERT INTO table\_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);

### 更新（Update）

* 更新特定行：

UPDATE table\_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;

### 删除（Delete）

* 删除特定行：

DELETE FROM table\_name  
WHERE condition;

# 4.实验内容

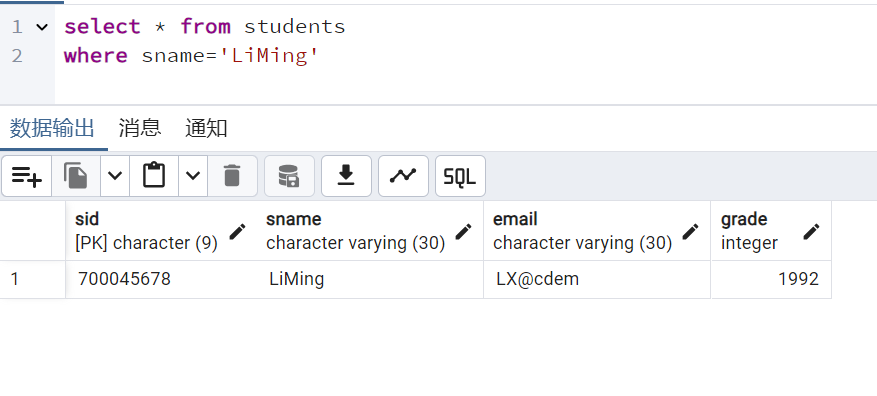
##### 1. 使用SQL语句向STUDENTS表中插入元组(编号：700045678；名字：LiMing；EMAIL：LX@cdemg.com；年级：1992)

insert into students(sid,sname,email,grade)  
values('700045678','LiMing','LX@cdemg.com',1992)

运行结果



查询语句执行

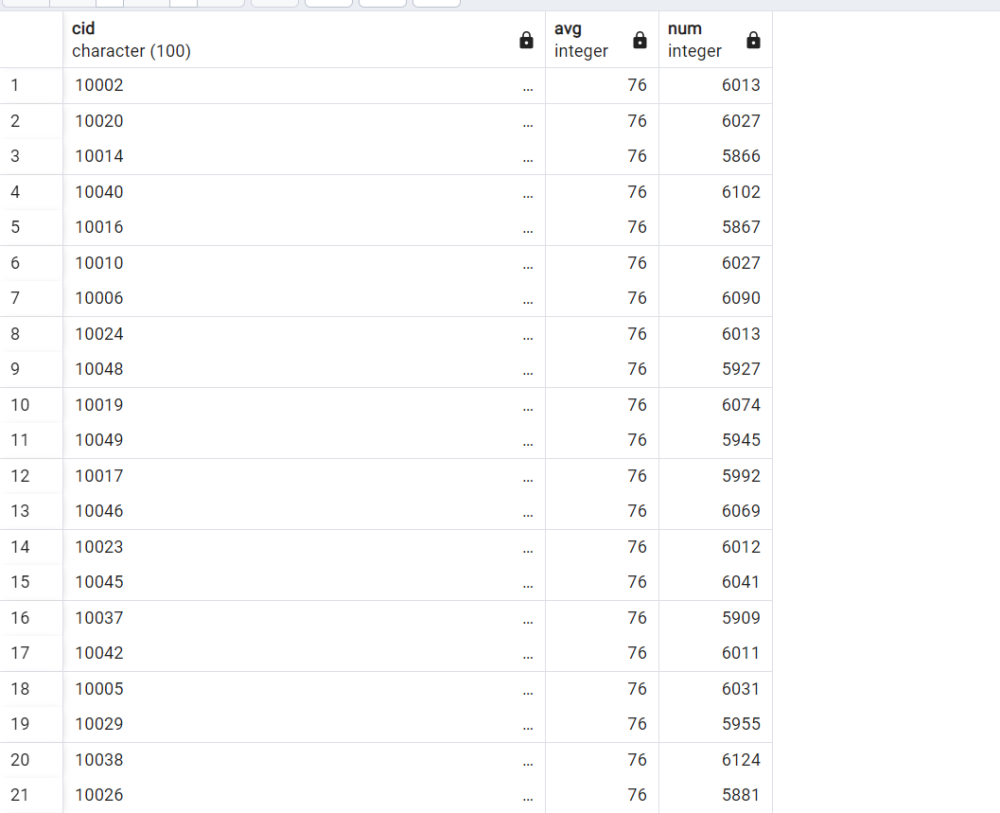


##### 2. 对每个课程，求学生的选课人数和学生的平均成绩，并把结果存入数据库。使用SELECT INTO和INSERT INTO两种方法实现。（提示：可先创建一个新表再插入数据）

create table courseinfo(  
 cid char(100),  
 avg int ,  
 num int  
);  
  
INSERT INTO courseinfo(cid,num,avg)  
SELECT cid,  
 COUNT(\*) AS num,  
 AVG(score) AS avg  
FROM choices  
GROUP BY cid;

创建新表，查询结果

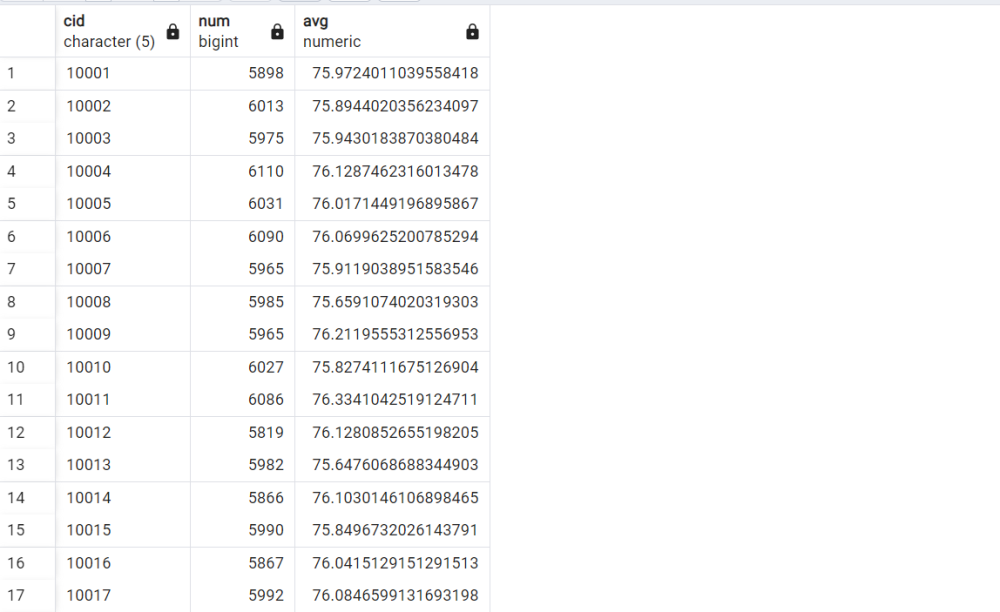
select \* from courseinfo



SELECT cid,  
 COUNT(\*) AS num,  
 AVG(score) AS avg  
into choiceinfo   
FROM choices  
GROUP BY cid;

查询结果

select \* from choiceinfo



##### 3. 在STUDENTS表中使用SQL语句将姓名为“LiMing”的学生的年级改为“2002”

update students  
set grade='2002'  
where sname='LiMing'

##### 4. 在TEACHERS表中使用SQL语句将所有教师的工资多加500元

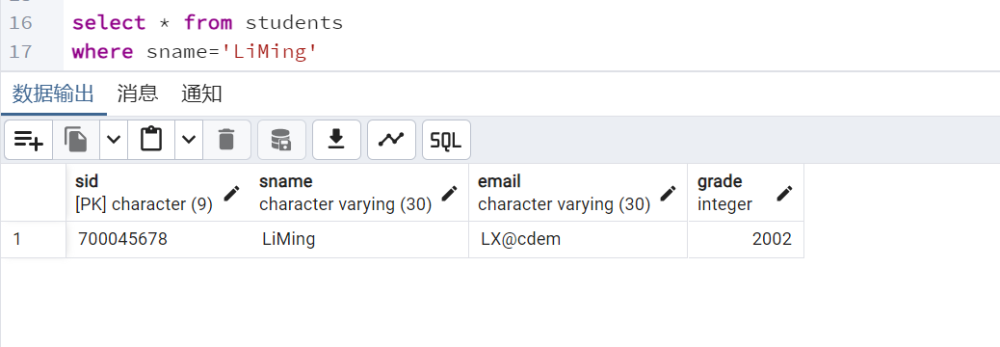
update teachers  
set salary=salary+500

##### 5. 将姓名为zapyv的学生的课程“C”的成绩加上5分

update choices   
set score=score+5  
where sid=  
(select sid from students where sname='zapyv')  
and  
cid=  
(select cid from courses where cname ='c')

##### 6. 在STUDENTS表中使用SQL语句删除姓名为“LiMing”的学生信息

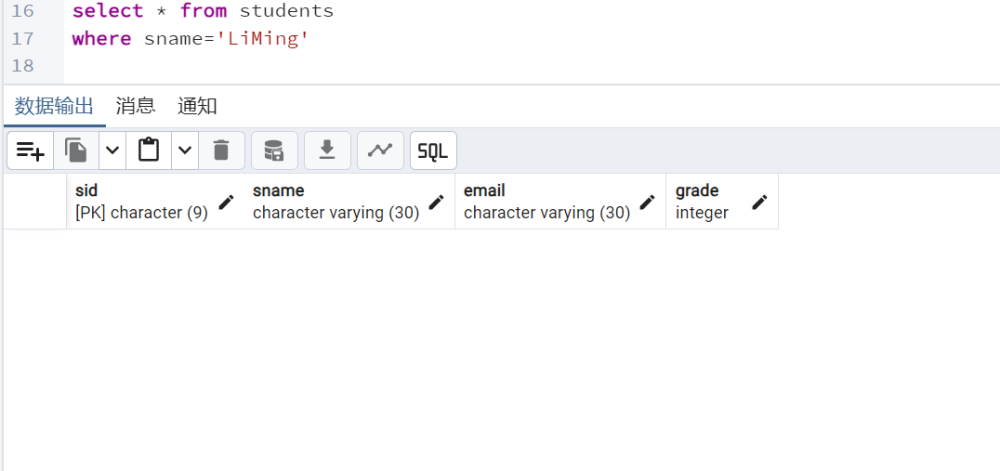
删除前



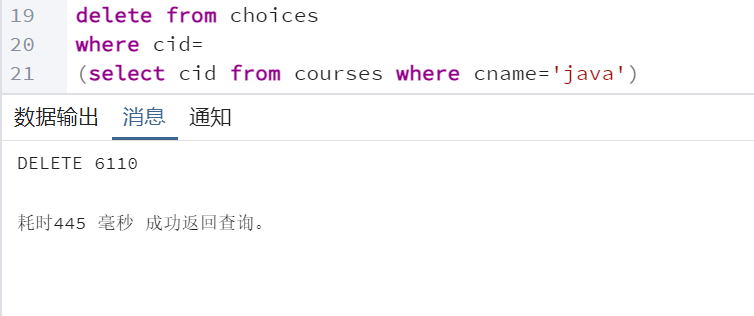
删除后

delete from students  
where sname='LiMing'

查询结果



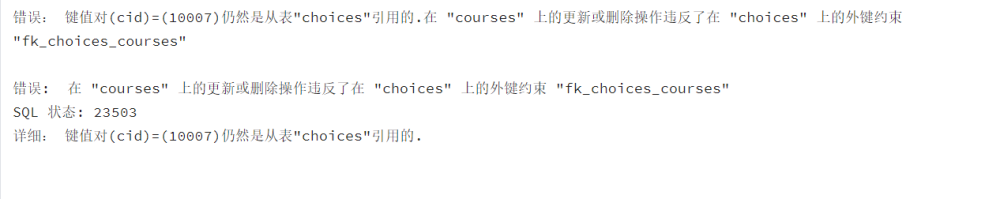
##### 7. 删除所有选修课程“Java”的选课记录



##### 8. 对COURSES表做删去时间<48的元组的操作，并讨论该删除操作所受到的约束

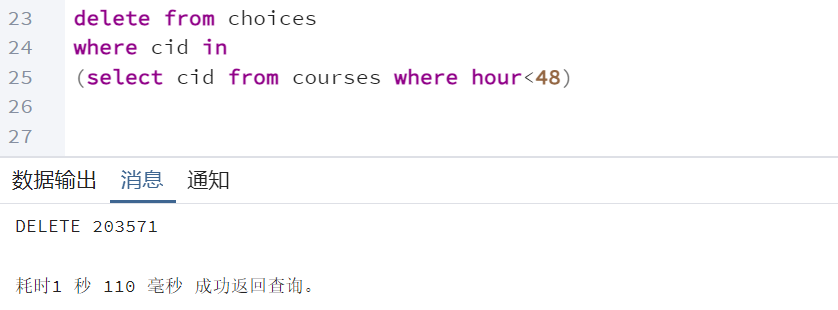
delete from courses   
where hour<48

会出现报错



其他表中存在依赖于 COURSES 表的外键，那么在删除 COURSES 表中的记录之前，需要先删除那些依赖的记录。否则，数据库会因为外键约束而拒绝删除操作。

更改操作为



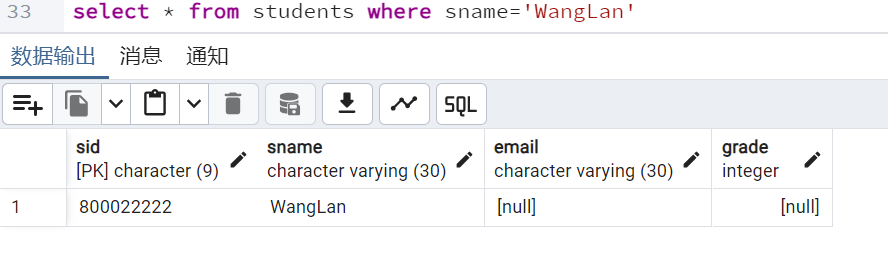
再次执行操作后成功删除

# 自我实践

##### 1. 向STUDENTS表插入编号是“800022222”且姓名是“WangLan”的元组

insert into students(sid,sname)  
values('800022222','WangLan')

插入后的查询结果

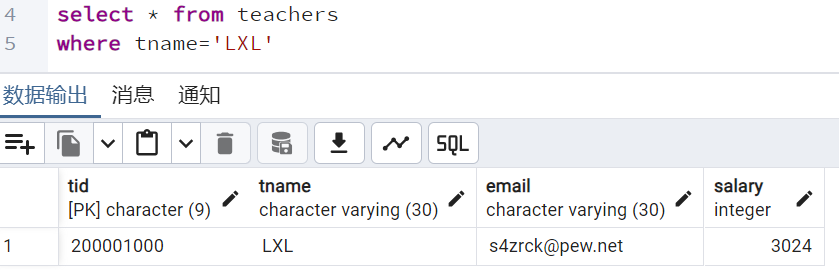


##### 2. 向TEACHERS表插入元组(“200001000”,"LXL",“s4zrck@pew.net”,“3024”)

insert into teachers(tid,tname,email,salary)  
values(“200001000”,"LXL",“s4zrck@pew.net”,“3024”)

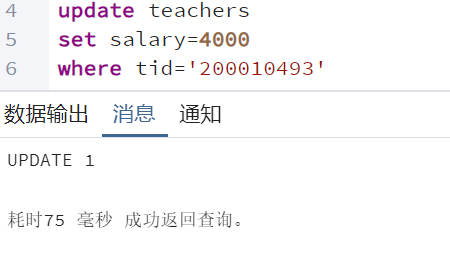


插入后的查询结果

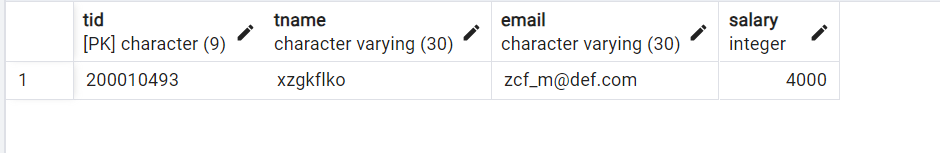


##### 3. 将TEACHERS表中编号为“200010493”的老师工资改为4000

update teachers  
set salary=4000  
where tid='200010493'

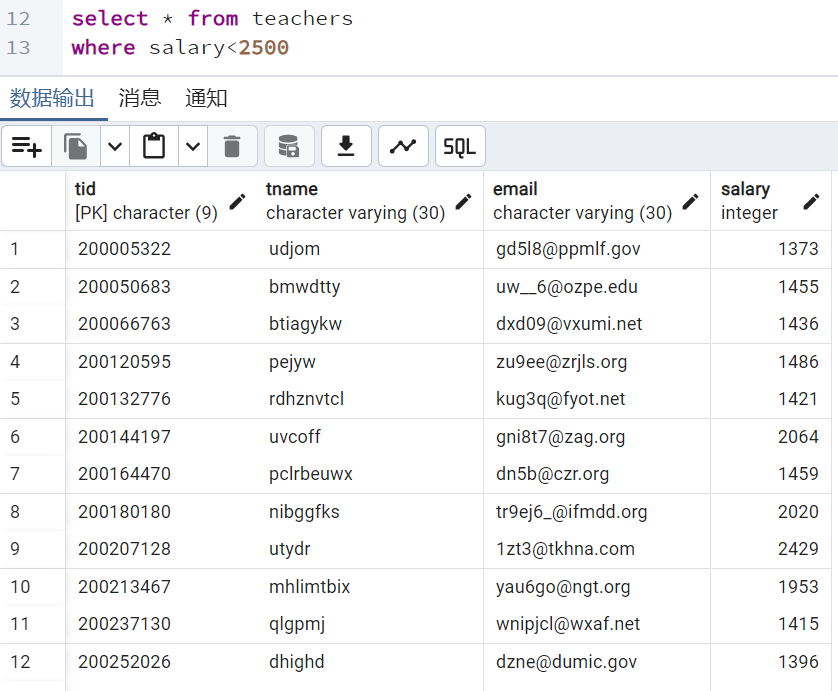


select \* from teachers  
where tid='200010493'



##### 4. 将TEACHERS表中所有工资小于2500的老师工资改为2500

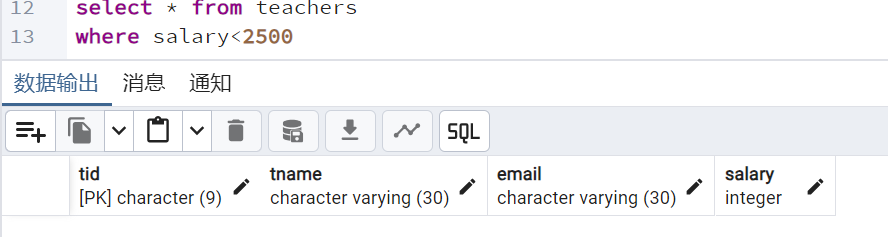
更改前



update teachers  
set salary=2500  
where salary<2500

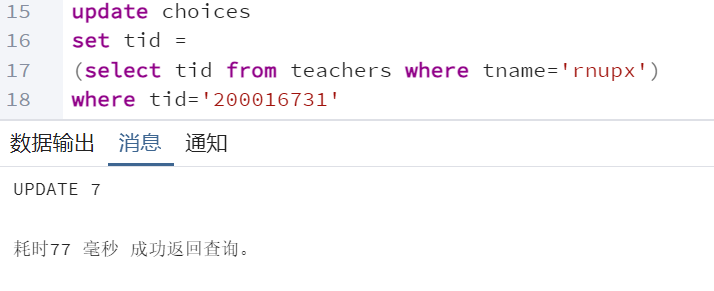


更改后



##### 5. 将由编号为“200016731”的老师讲授的课程全部改成由姓名为“rnupx”的老师讲授

update choices   
set tid =  
(select tid from teachers where tname='rnupx')  
where tid='200016731'



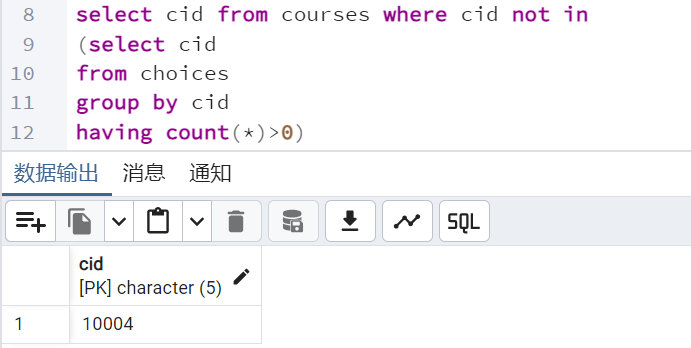
##### 6. 更新编号“800071780”的学生年级为“2001”

update students  
grade=2001  
where sid='800071780'



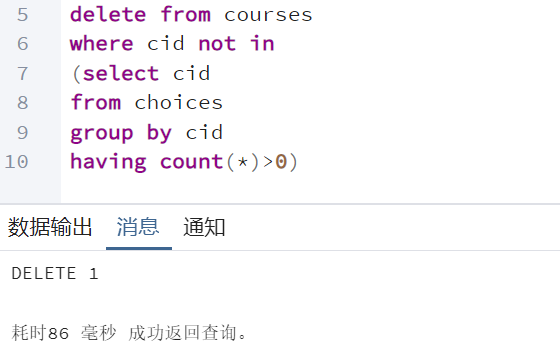
##### 7. 删除没有学生选修的课程

存在课程没有学生选择



删除代码

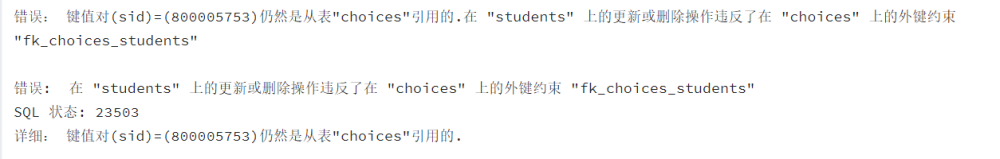
delete from courses  
where cid not in  
(select cid   
from choices   
group by cid  
having count(\*)>0)



##### 8. 删除年级高于1998的学生信息

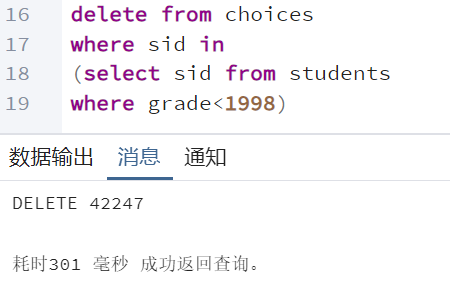
delete from students   
where  
grade<1998

其他表中存在依赖于 studentS 表的外键，那么在删除 studentS 表中的记录之前，需要先删除那些依赖的记录。否则，数据库会因为外键约束而拒绝删除操作。

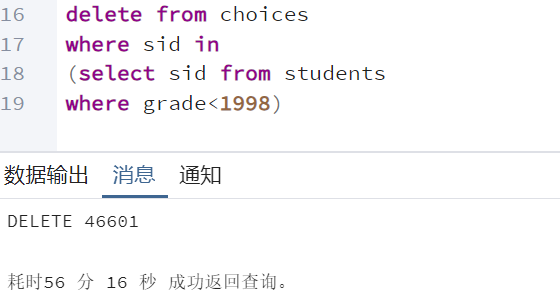


更改操作为

delete from choices   
where sid in   
(select sid from students   
where grade<1998)



##### 9. 删除没有选修课程的学生信息

耗时较长，因为复杂度去到O(n\*\*2)

##### 10. 删除成绩不及格的选课记录

delete from choices  
where score<60

# 实验问题

### 注意事项

* 在执行更新和删除操作时，务必小心使用 WHERE 子句，以确保只修改或删除预期的行。否则操作很难撤回，对表造成不可逆的伤害操作

* 在执行插入操作时，确保插入的数据符合表的约束，如数据类型、非空约束等。

* 在执行删除操作时，考虑到外键约束，可能需要先删除或更新相关的子表数据。否则可能出现奇怪报错

例如