实验五 数据更新

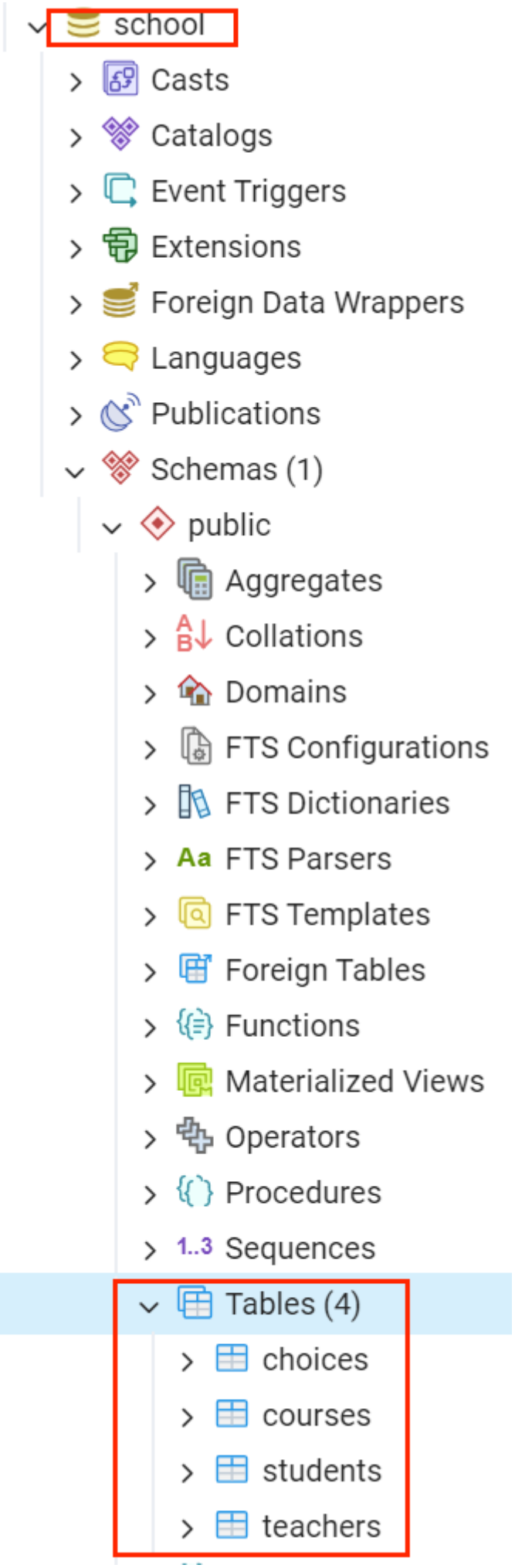
【姓名】谢宇桐 【学号】 22336259 【专业】计算机科学与技术

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

已安装完成Postgre SQL，在pgAdmin 4进行编辑。并已配置好schoo数据库的四张表格。


school
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas (1)
public
Aggregates
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables (4)
choices
courses
students
teachers

3.实验内容

在本次实验中，主要的内容是如何使用 SQL 语句对数据进行更新。

本节实验的主要内容包括：

- 使用INSERT INTO语句插入数据，包括插入一个元组或将子查询的结果插入到数据库中两种方式。
- 使用SELECT INTO语句，产生一个新表并插入数据。
- 使用UPDATE 语句可以修改指定表中满足WHERE子句条件的元组，有三种修改的方式：修改某一个元组的值，修改多个元组的值，带子查询的修改语句。
- 使用DELETE 语句删除数据：删除某一个元组的值，删除多个元组的值，带子查询的删除语句。
  注："SELECT INTO"是SQL查询语句的一部分，用于将查询的结果插入到新表中。它的语法如下：

```
SELECT column1, column2, ...
INTO new_table
FROM existing_table
WHERE condition;
```

其中：

column1, column2, ... 是要选择的列。

new_table 是要将结果插入的新表。

existing_table 是要从中选择数据的现有表。

condition 是可选的筛选条件。

这个语句执行时，它会从现有表中选择指定列的数据，并将结果插入到新表中。

4.实验步骤（遇到的问题为灰色字）

在数据库中，存在这样的关系：学生可以选择课程。一个课程对应一个教师。在表CHOICES中保存学生的选课记录。

STUDENTS(sid,sname,email,grade)

TEACHERS(tid,tname,email,salary)

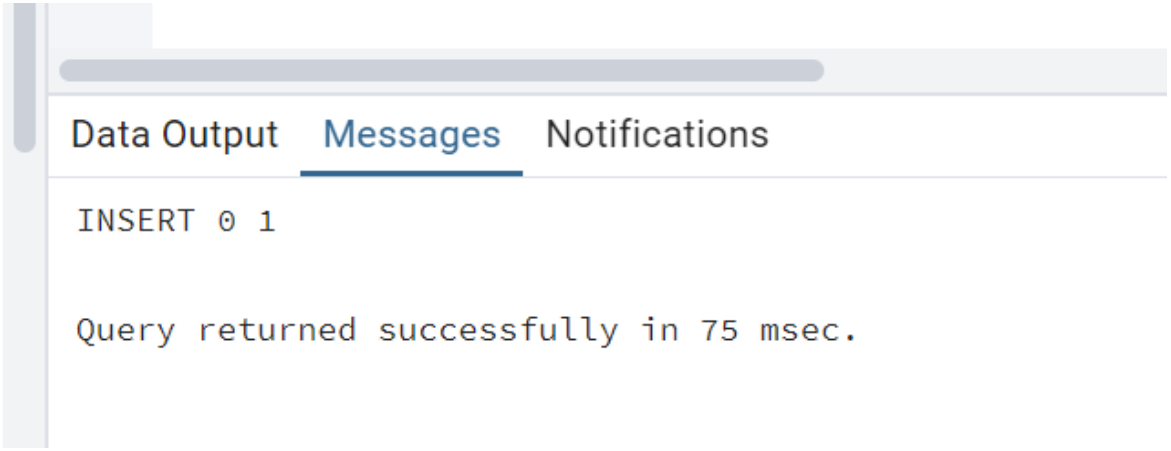
COURSES(cid,cname,hour)

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

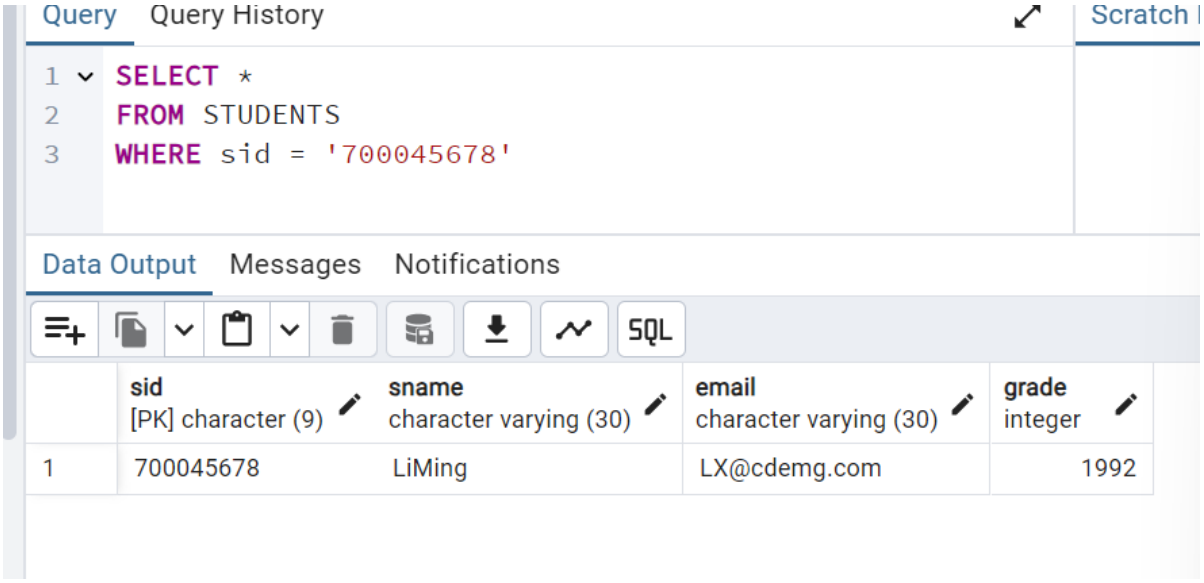
在数据库school中按下列要求进行数据更新。

(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');
```



验证一下：



可见插入成功

(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）；

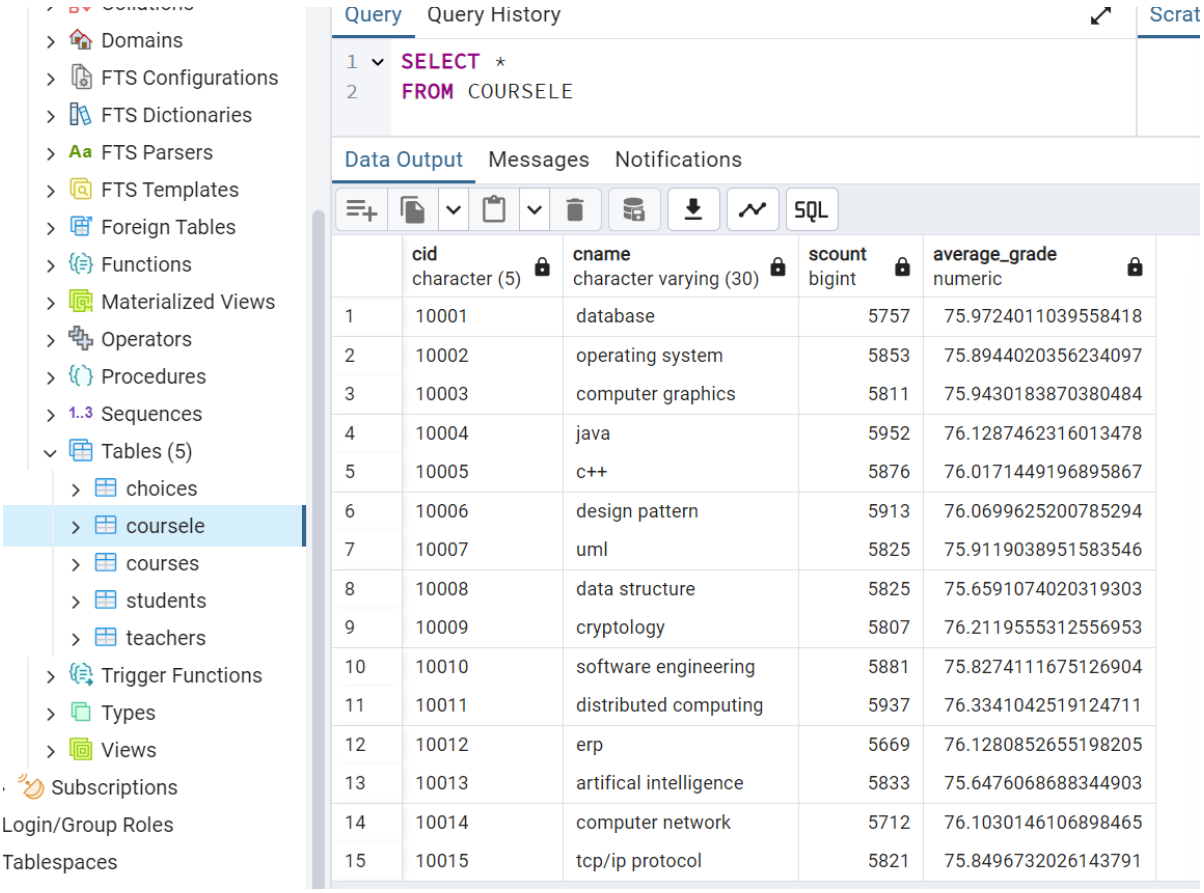
SELECT INTO 用于从现有的表中选择数据并创建一个新表，同时将查询结果插入到这个新表中。

INSERT INTO用于从现有的表中选择数据将查询结果插入到现有的表中。

SELECT INTO：

```
SELECT COURSES.cid,
       COURSES.cname,
       COUNT(DISTINCT CHOICES.sid) AS scount,
       AVG(CHOICES.score) AS average_grade
INTO COURSELE
FROM CHOICES
JOIN COURSES ON CHOICES.cid = COURSES.cid
GROUP BY COURSES.cid, COURSES.cname;
```

验证一下：

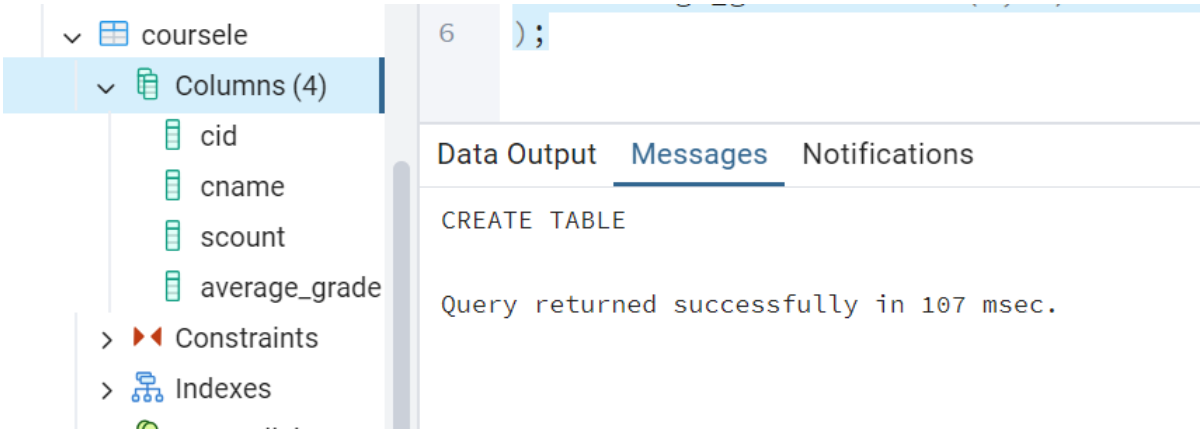


我们看到创建成功。

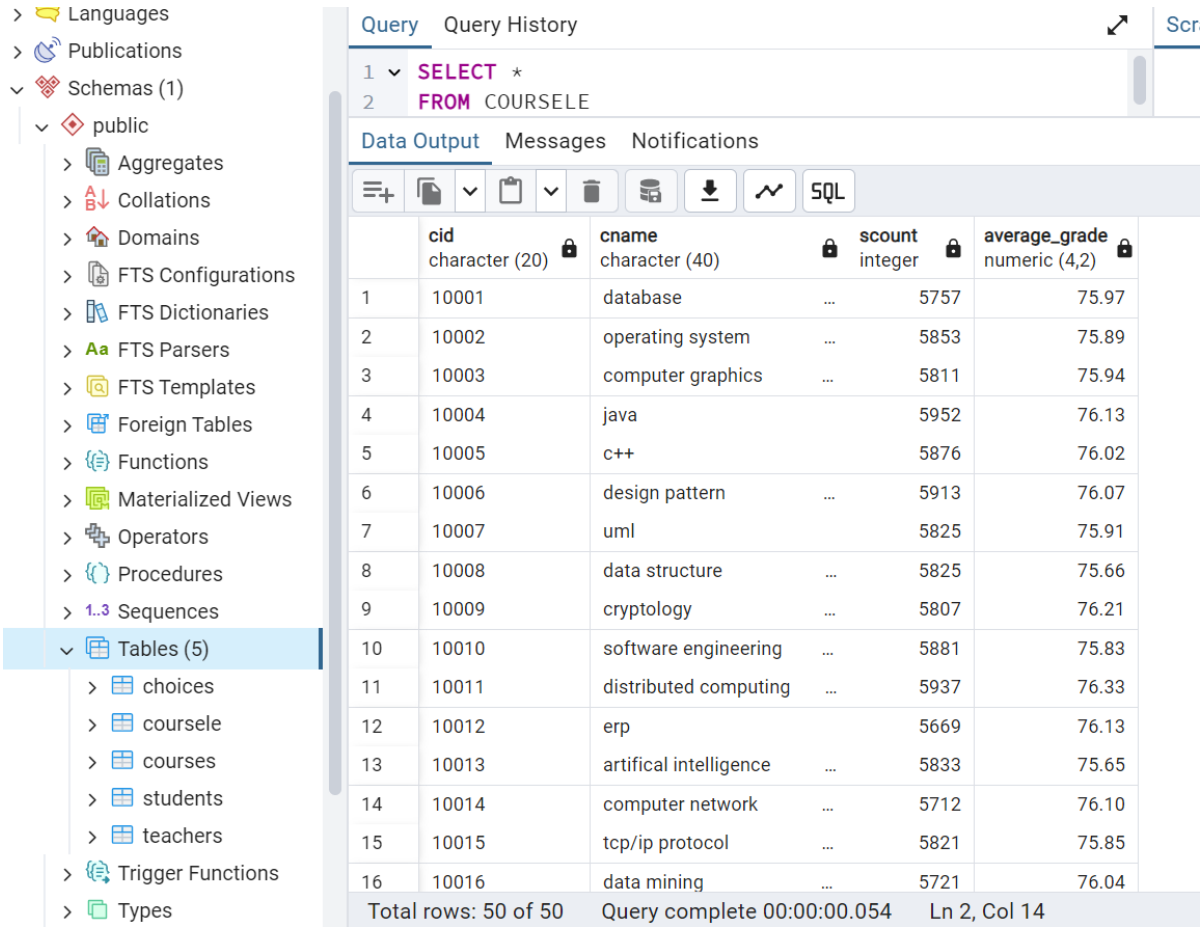
INSERT INTO:

我们先把上面写好的COURSELE删掉。然后创建一个新表名为COURSELE:

```
CREATE TABLE COURSELE (
    cid CHAR(20),
    cname CHAR(40),
    scount INT,
    average_grade DECIMAL(4, 2)-- DECIMAL(4, 2)表示这是一个小数类型的字段，其中4指定了数字总体有4位，2指定了小数点后有两位。
);
```



```sql
INSERT INTO COURSELE (cid, cname, scount, average_grade)
SELECT COURSES.cid,
       COURSES.cname,
       COUNT(DISTINCT CHOICES.sid) AS scount,
       AVG(CHOICES.score) AS average_grade
FROM CHOICES
JOIN COURSES ON CHOICES.cid = COURSES.cid
GROUP BY COURSES.cid, COURSES.cname;
```



可以看到结果和SELECT一致，创建完成

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

```sql
UPDATE STUDENTS
SET grade = '2002'
WHERE sname = 'LiMing';
```

Query　Query History

```
1 UPDATE STUDENTS
2 SET grade = '2002'
3 WHERE sname = 'LiMing';
```

Data Output　Messages　Notifications

```
UPDATE 1

Query returned successfully in 123 msec.
```

验证看到修改完毕:

```
1 SELECT sname, grade
2 FROM STUDENTS
3 WHERE sname = 'LiMing'
```

Data Output　Messages　Notifications

	sname character varying (30)	grade integer
1	LiMing	2002

(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;--这句会使工资为NULL的老师仍为NULL
SET salary = COALESCE(salary, 0) + 500;--这句会使工资为NULL的老师也+500
```

Query　Query History

```
1 UPDATE TEACHERS
2 SET salary = salary + 500;
```

Data Output　Messages　Notifications

```
UPDATE 15000

Query returned successfully in 354 msec.
```

验证可得加工资成功

```
1 SELECT *
2 FROM TEACHERS
```

Data Output　Messages　Notifications

	tid [PK] character (9)	tname character varying (30)	email character varying (30)	salary integer
1	200128004	ztvosbydu	_b53@ygzs.edu	4862
2	200132776	rdhznvtcl	kug3q@fyot.net	1421
3	200144197	uvcoff	gni8t7@zag.org	2064
4	200148844	hqmwjp	gjm17@txcex.edu	4017
5	200153917	tooaa	99bag1q@idg.org	3558
6	200159745	kmnohpaj	_ogqj@heh.net	3437
7	200164470	pclrbeuwx	dn5b@czr.org	1459
8	200166655	tcufyzefp	96ze7in@kkwct.edu	4433
9	200169488	abtqstb	56hf@wsvf.net	4389
10	200171064	ptmzib	fmr9co7@fjtk.org	2671
11	200180180	nibggfks	tr9ej6_@ifmdd.org	2020
12	200202973	gdgcxjhj	3529@knnbo.edu	3638

(5) 将姓名为zapyv的学生的课程“C”的成绩加上5分;

他原成绩为:

```
SELECT CHOICES.score
FROM STUDENTS
JOIN CHOICES ON STUDENTS.sid = CHOICES.sid
JOIN COURSES ON CHOICES.cid = COURSES.cid
WHERE STUDENTS.sname = 'zapyv' AND COURSES.cname = 'c';
```

Data Output | Messages | Notifications

	score integer
1	80

```
UPDATE CHOICES
SET score = score + 5
WHERE sid IN (
    SELECT sid
    FROM STUDENTS
    WHERE sname = 'zapyv'
)
AND cid IN (
    SELECT cid
    FROM COURSES
    WHERE cname = 'c'
);
```

```
SELECT CHOICES.score
FROM STUDENTS
JOIN CHOICES ON STUDENTS.sid = CHOICES.sid
JOIN COURSES ON CHOICES.cid = COURSES.cid
WHERE STUDENTS.sname = 'zapyv' AND COURSES.cname = 'c';
```

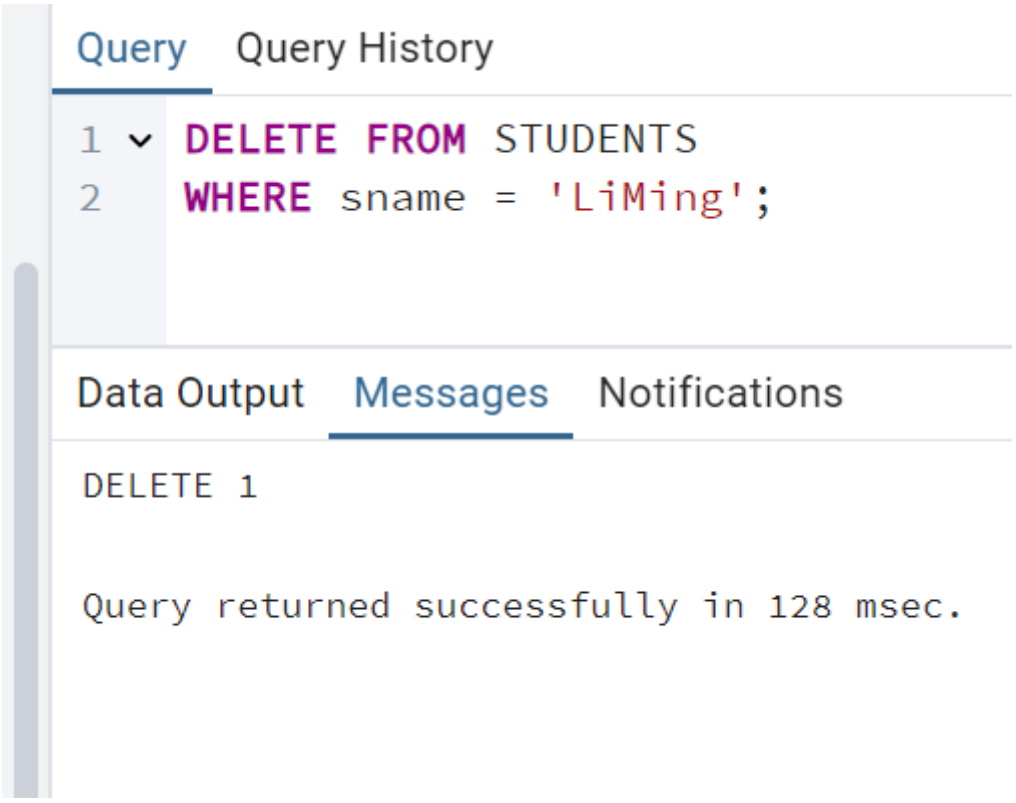
Data Output | Messages | Notifications

	score integer
1	85

可以看到成功加了5分

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

```
DELETE FROM STUDENTS
WHERE sname = 'LiMing';
```



验证可以看到已经找不到LiMing的信息了:

(7) 删除所有选修课程“Java”的选课记录;

```
DELETE FROM CHOICES
WHERE cid IN (
    SELECT cid
    FROM COURSES
    WHERE cname = 'java'
);
```

Query Query History

```
1  DELETE FROM CHOICES
2  WHERE cid IN (
3      SELECT cid
4      FROM COURSES
5      WHERE cname = 'java'
6  );
```

Data Output Messages Notifications

```
DELETE 6110

Query returned successfully in 520 msec.
```

同样的，也找不到关于java的所有信息了

Query Query History Scra

```
1  SELECT *
2  FROM CHOICES
3  WHERE cid IN (
4      SELECT cid
5      FROM COURSES
6      WHERE cname = 'java'
7  );
```

Data Output Messages Notifications

SQL

	no [PK] integer	sid character (9)	tid character (9)	cid character (5)	score integer

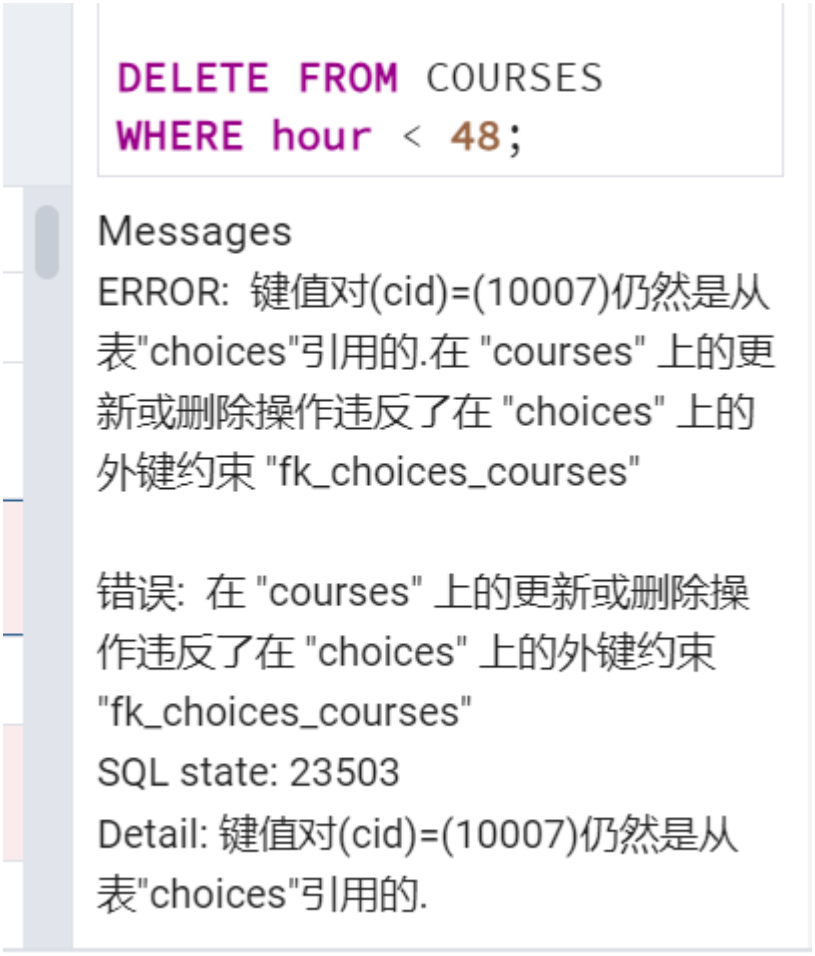
(8) 对COURSES表做删去时间<48的元组的操作，并讨论该删除操作所受到的约束。

```
-- 先删除引用了这些课程的选课记录
DELETE FROM CHOICES
WHERE cid IN (
    SELECT cid
    FROM COURSES
    WHERE hour < 48
);

-- 然后删除课程记录
DELETE FROM COURSES
WHERE hour < 48;
```

这个操作受到外键约束（它确保在一个表中的外键字段的值必须在另一个表的对应字段中有对应的值）的影响，因为 CHOICES 表引用了 COURSES 表中的 cid。如果 CHOICES 表设置了级联删除，则不需要额外操作；否则，需要先删除或更新引用这些 cid 的记录。

（报错：

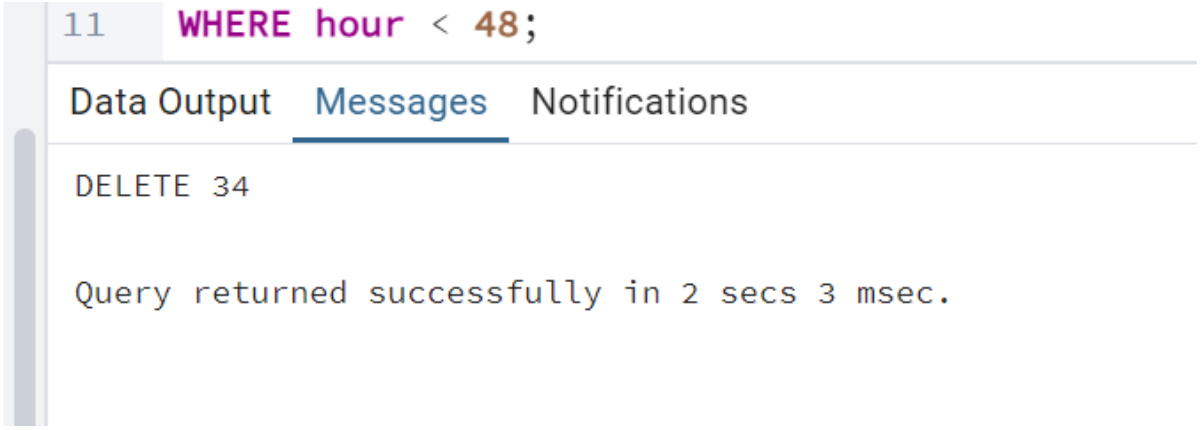


级联删除与创建：

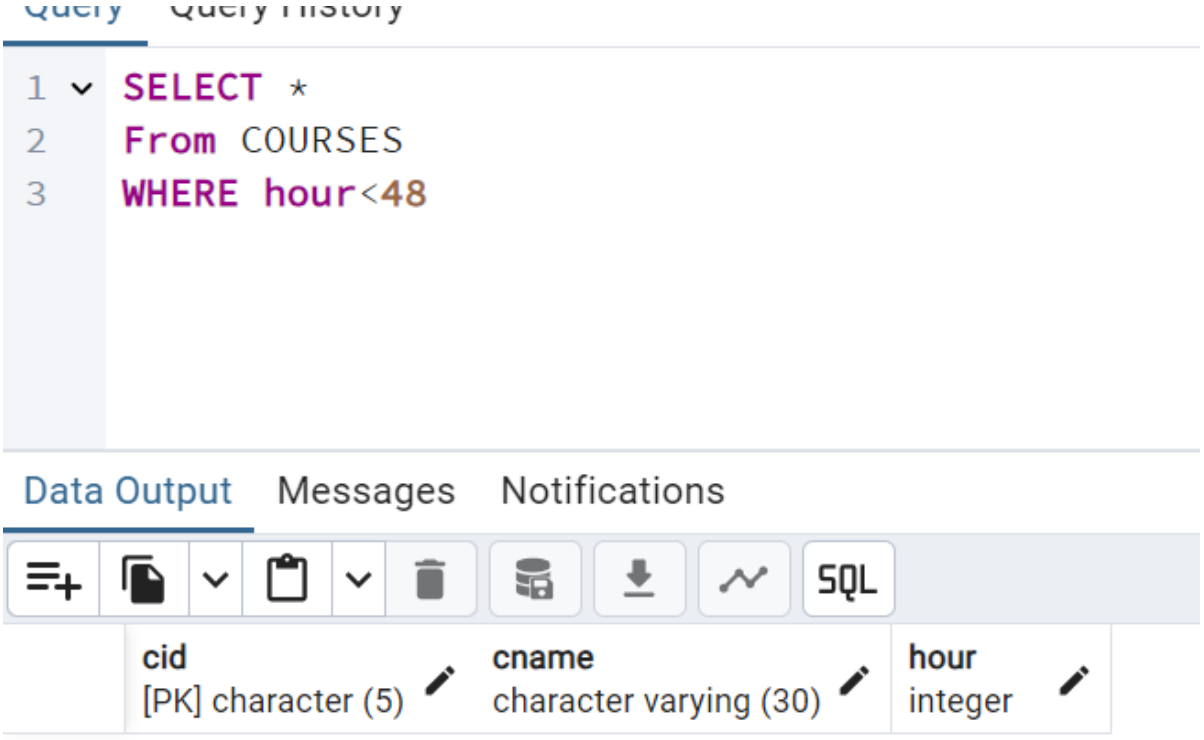
```sql
--删除级联
ALTER TABLE CHOICES
DROP CONSTRAINT fk_CHOICES_COURSES;

--创建级联
ALTER TABLE CHOICES
ADD CONSTRAINT fk_CHOICES_COURSES
FOREIGN KEY (cid)
REFERENCES COURSES(cid)
ON DELETE CASCADE;
```

但本题未运用级联删除，直接按照最上面的代码先进行删除）



我们可以看到已经找不到课程时长小于48的课程了

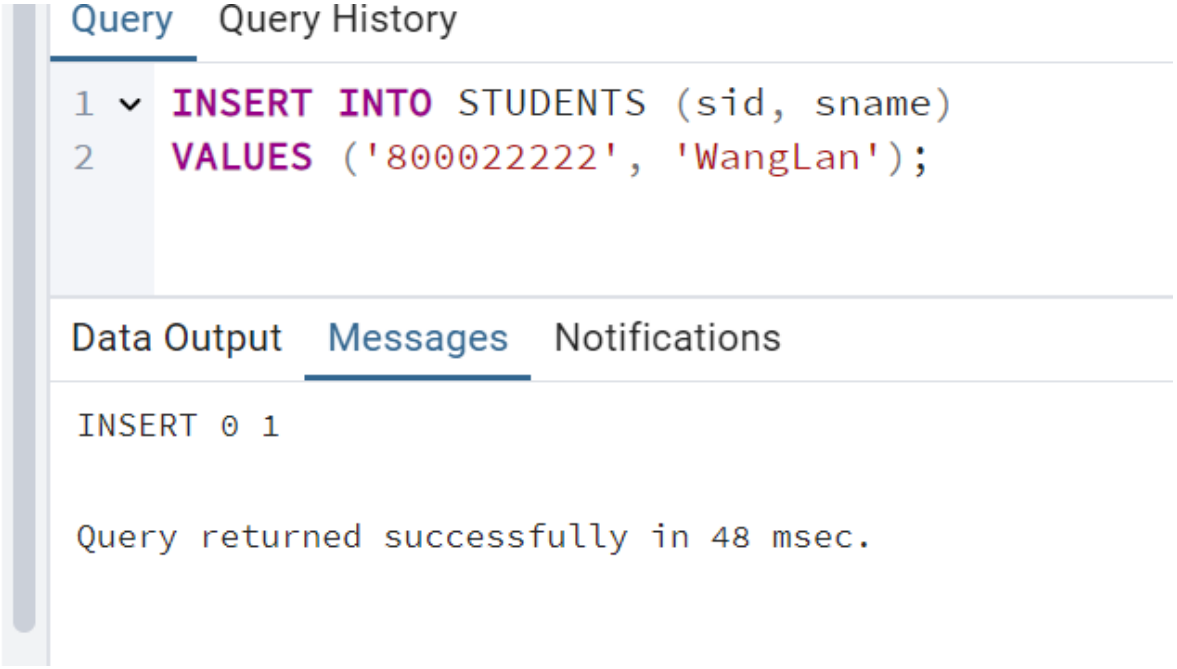


至此，课内实验成功完成。

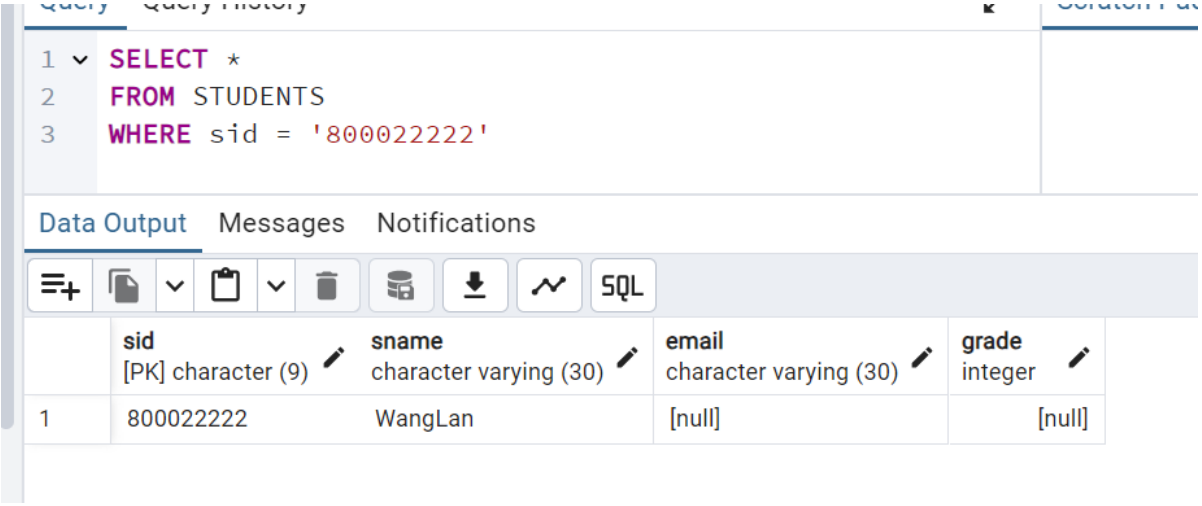
5.“自我实践”实验步骤

(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');
```



经过验证查询得知插入成功



(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);
```

Query　Query History

```
1 INSERT INTO TEACHERS (tid, tname, email, salary)
2 VALUES ('200001000', 'LXL', 's4zrck@pew.net', 3024);
```

Data Output　Messages　Notifications

```
INSERT 0 1

Query returned successfully in 59 msec.
```

经过验证得插入成功：

```
1 SELECT *
2 FROM TEACHERS
3 WHERE tid = '200001000'
```

Data Output　Messages　Notifications

	tid [PK] character (9)	tname character varying (30)	email character varying (30)	salary integer
1	200001000	LXL	s4zrck@pew.net	3024

(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';
```

Query　Query History

```
1  UPDATE TEACHERS
2  SET salary = 4000
3  WHERE tid = '200010493';
```

Data Output　Messages　Notifications

```
UPDATE 1

Query returned successfully in 65 msec.
```

经过查询验证可得修改成功：

Query　Query History　Scratch Pa

```
1  SELECT *
2  FROM TEACHERS
3  WHERE tid = '200010493'
```

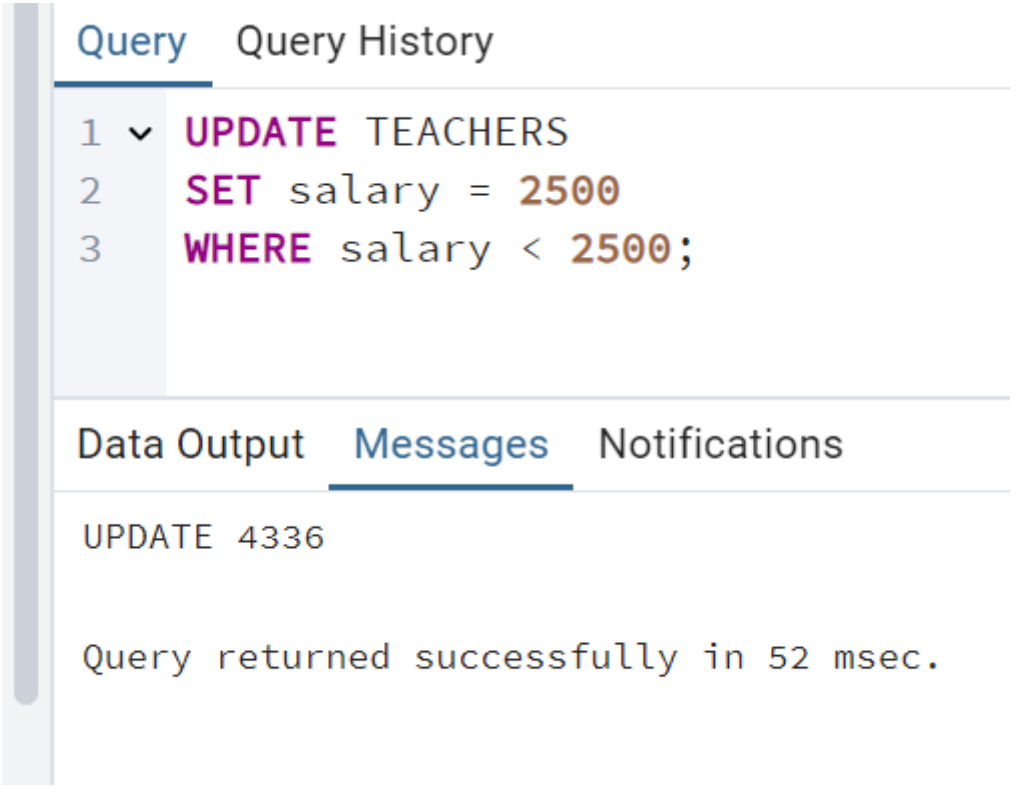
Data Output　Messages　Notifications

SQL

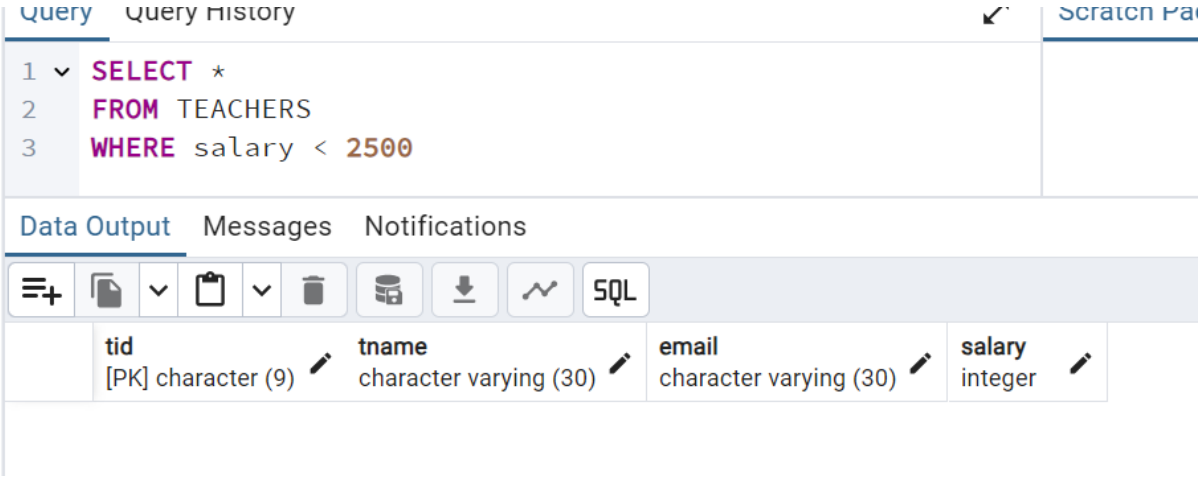
	tid [PK] character (9)	tname character varying (30)	email character varying (30)	salary integer
1	200010493	xzgkflko	zcf_m@def.com	4000

(4) 将TEACHERS表中所有工资小于2500的老师工资改为2500;

```
UPDATE TEACHERS
SET salary = 2500
WHERE salary < 2500;
```



经过查询验证可得表内已没有小于2500工资的老师：



(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;

我们先查一下修改前的情况，方便验证：

Query Query History

```
1 SELECT *
2 FROM CHOICES
3 WHERE tid = '200016731'
```

Data Output Messages Notifications

	no [PK] integer	sid character (9)	tid character (9)	cid character (5)	score integer
1	533668489	860975663	200016731	10008	90
2	534705108	889135469	200016731	10029	74
3	553313919	807673623	200016731	10005	81
4	556542980	843706221	200016731	10001	96
5	567187005	805064362	200016731	10003	62
6	570672378	834231097	200016731	10017	76
7	591347658	883565121	200016731	10014	79

可以看到tid为200016731的老师教授七门课

```
UPDATE CHOICES
SET tid = (
    SELECT tid
    FROM TEACHERS
    WHERE tname = 'rnupx'
)
WHERE tid = '200016731';
```

```sql
UPDATE CHOICES
SET tid = (
    SELECT tid
    FROM TEACHERS
    WHERE tname = 'rnupx'
)
WHERE tid = '200016731';
```

Data Output | Messages | Notifications

```
UPDATE 7

Query returned successfully in 54 msec.
```

修改后我们已经查不到200016731老师讲的课，说明修改成功：

```sql
SELECT *
FROM CHOICES
WHERE tid = '200016731'
```

Data Output | Messages | Notifications

no [PK] integer	sid character (9)	tid character (9)	cid character (5)	score integer

(6) 更新编号“800071780”的学生年级为“2001”;

```sql
UPDATE STUDENTS
SET grade = '2001'
WHERE sid = '800071780';
```

Query Query History

```
1 UPDATE STUDENTS
2 SET grade = '2001'
3 WHERE sid = '800071780';
```

Data Output Messages Notifications

```
UPDATE 1

Query returned successfully in 95 msec.
```

验证修改成功：

Query Query History Scratch

```
1 SELECT *
2 FROM STUDENTS
3 WHERE sid = '800071780'
```

Data Output Messages Notifications

	sid [PK] character (9)	sname character varying (30)	email character varying (30)	grade integer
1	800071780	iscmwink	htoactp@aof.gov	2001

(7) 删除没有学生选修的课程;

```
DELETE FROM COURSES
WHERE cid NOT IN (
    SELECT cid
    FROM CHOICES
);
```

```
DELETE FROM COURSES
WHERE cid NOT IN (
    SELECT cid
    FROM CHOICES
);
```

Data Output | Messages | Notifications

```
DELETE 1

Query returned successfully in 60 msec.
```

查询不到没有学生选的课程说明删除成功：

```
SELECT *
FROM COURSES
WHERE cid NOT IN (
    SELECT cid
    FROM CHOICES
);
```

Data Output | Messages | Notifications

cid [PK] character (5)	cname character varying (30)	hour integer

(8) 删除年级高于1998的学生信息;

本题一样会运用到级联删除与创建，但为了后续做题方便，这里不进行级联删除这一步操作，但我将代码都附上来，如下：

先删除引用记录：

```sql
-- 首先删除CHOICES表中引用的记录
DELETE FROM CHOICES
WHERE sid IN (
    SELECT sid
    FROM STUDENTS
    WHERE grade > '1998'
);

-- 然后删除STUDENTS表中的记录
DELETE FROM STUDENTS
WHERE grade > '1998';
```

先删除级联:

```sql
-- 移除现有的外键约束
ALTER TABLE CHOICES DROP CONSTRAINT fk_CHOICES_STUDENTS;

-- 然后删除STUDENTS表中的记录
DELETE FROM STUDENTS
WHERE grade > '1998';
```

Query Query History

```sql
-- 移除现有的外键约束
ALTER TABLE CHOICES
DROP CONSTRAINT fk_CHOICES_STUDENTS;

-- 然后删除STUDENTS表中的记录
DELETE FROM STUDENTS
WHERE grade > '1998';
```

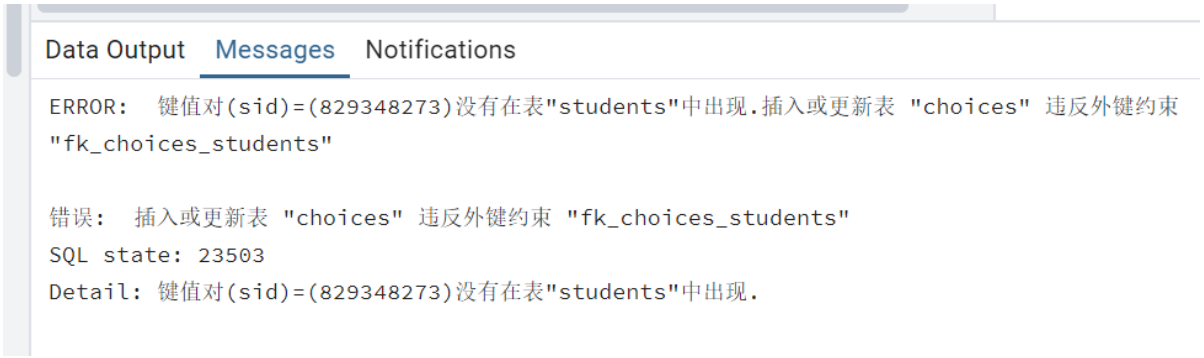
Data Output Messages Notifications

```
DELETE 40092

Query returned successfully in 53 msec.
```

这里有一个问题，就是当我删除了STUDENTS表的记录，再想按照下面方法恢复级联，就会出现报错：

```
-- 重新创建外键约束，并添加ON DELETE CASCADE选项，即如果在 STUDENTS 表中删除了一个学生记录，那么在 CHOICES 表中所有 sid 与被删除学生 sid 相匹配的行也会被自动删除。
ALTER TABLE CHOICES
ADD CONSTRAINT fk_CHOICES_STUDENTS
FOREIGN KEY (sid) REFERENCES STUDENTS(sid) ON DELETE CASCADE;
```



即两者的sid并不完全匹配，因此创建不了级联。

在这里我想到的解决方法是把CHOICES里对应STUDENTS里不存在的sid的学生删掉，保持两表的统一，再进行恢复。我将剩下的实验做完后，感觉恢复太麻烦，就使用课内实验(8)的fk_CHOICES_COURSES条件进行试验：

首先两段代码完全正确：

```
--删除级联
ALTER TABLE CHOICES
DROP CONSTRAINT fk_CHOICES_COURSES;

--创建级联
ALTER TABLE CHOICES
ADD CONSTRAINT fk_CHOICES_COURSES
FOREIGN KEY (cid)
REFERENCES COURSES(cid)
ON DELETE CASCADE;
```

可以直接运行。且只能通过这种删除外键约束再创建的方法设置ON DELETE CASCADE属性，而不能在有级联的基础上直接进行设置。

此时再运行下面代码，就不会报错了：

Query Query History

```
1 DELETE FROM COURSES
2 WHERE hour < 48;
3
```

Data Output Messages Notifications

```
DELETE 0

Query returned successfully in 46 msec.
```

(9) 删除没有选修课程的学生信息;

因为上题把级联删除了，我们直接删除对应信息即可

```
DELETE FROM STUDENTS
WHERE sid NOT IN (
    SELECT sid
    FROM CHOICES
);
```

Query Query History

```
1 DELETE FROM STUDENTS
2 WHERE sid NOT IN (
3     SELECT sid
4     FROM CHOICES
5 );
```

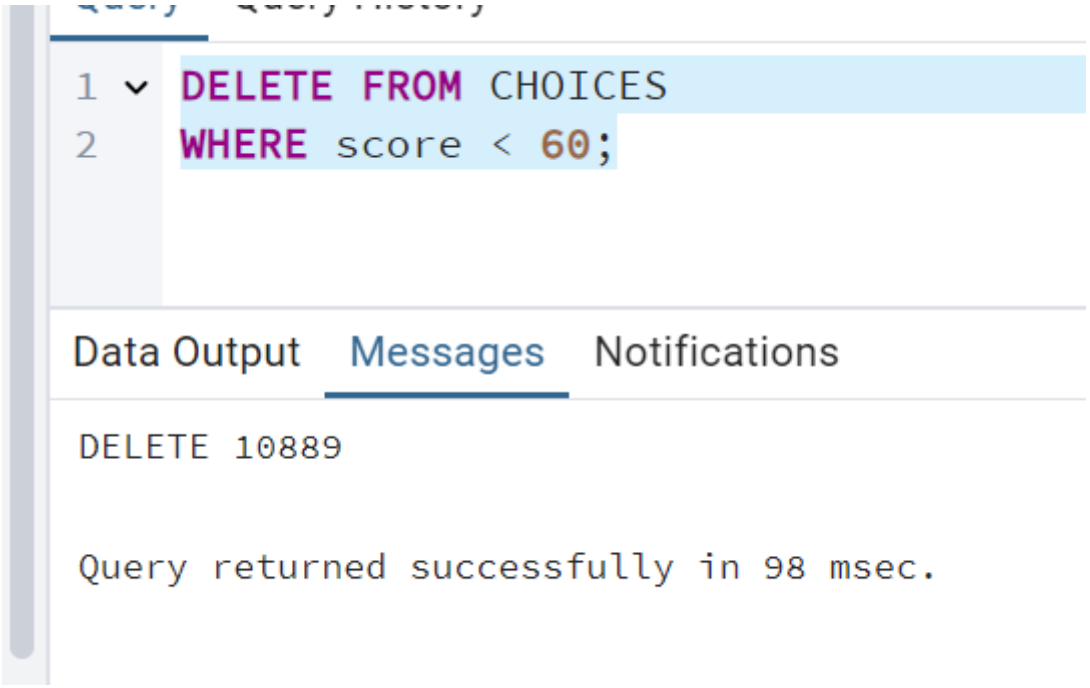
Data Output Messages Notifications

```
DELETE 23187

Query returned successfully in 88 msec.
```

(10) 删除成绩不及格的选课记录。

```
DELETE FROM CHOICES
WHERE score < 60;
```



验证可得删除成功：