

# 数据库实验报告

---

## 1. 实验目的

## 2. 实验环境

## 3. 实验常用语句

插入 (Insert)

更新 (Update)

删除 (Delete)

## 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@cdem g....
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 INSE...
3. 在STUDENTS表中 使用SQL语句将姓名为“LiMing”的学生的年级改为“2002”
4. 在TEACHERS表中 使用SQL语句将所有教师的工资多加500元
5. 将姓名为zapyv的学生的课程“C”的成绩加上5分
6. 在STUDENTS表中 使用SQL语句删除姓名为“LiMing”的学生信息
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

## 自我实践

1. 向STUDENTS表插入编号是“800022222”且姓名是“WangLan”的元组
2. 向TEACHERS表插入元组("200001000","LXL","s4zrck@pew.net","3024")
3. 将TEACHERS表中编号为“200010493”的老师工资改为4000
4. 将TEACHERS表中所有工资小于2500的老师工资改为2500
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
6. 更新编号“800071780”的学生年级为“2001”
7. 删除没有学生选修的课程
8. 删除年级高于1998的学生信息
9. 删除没有选修课程的学生信息
10. 删除成绩不及格的选课记录

## 实验问题

### 注意事项

# 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：

```
1  INSERT INTO table_name (column1, column2, column3, ...)
2  VALUES (value1, value2, value3, ...);
```

### 更新 (Update)

- 更新特定行：

```
1 UPDATE table_name
2 SET column1 = value1, column2 = value2, ...
3 WHERE condition;
```

## 删除 (Delete)

- 删除特定行：

```
1 DELETE FROM table_name
2 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)

▼ Python |

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

### 运行结果

INSERT 0 1

耗时54 毫秒 成功返回查询。

### 查询语句执行

```

1 select * from students
2 where sname='LiMing'

```

数据输出 消息 通知

	sid [PK] character (9)	sname character varying (30)	email character varying (30)	grade integer
1	700045678	LiMing	LX@cdem	1992

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）

insert into实现

Python

```

1 create table courseinfo(
2     cid char(100),
3     avg int ,
4     num int
5 );
6
7 INSERT INTO courseinfo(cid,num,avg)
8 SELECT cid,
9         COUNT(*) AS num,
10        AVG(score) AS avg
11 FROM choices
12 GROUP BY cid;
13

```

创建新表，查询结果

```

1
2 select * from courseinfo

```

	cid character (100)	avg integer	num integer
1	10002	...	76
2	10020	...	76
3	10014	...	76
4	10040	...	76
5	10016	...	76
6	10010	...	76
7	10006	...	76
8	10024	...	76
9	10048	...	76
10	10019	...	76
11	10049	...	76
12	10017	...	76
13	10046	...	76
14	10023	...	76
15	10045	...	76
16	10037	...	76
17	10042	...	76
18	10005	...	76
19	10029	...	76
20	10038	...	76
21	10026	...	76

▼ select into实现

Python |

```

1  SELECT cid,
2      COUNT(*) AS num,
3      AVG(score) AS avg
4  into choiceinfo
5  FROM choices
6  GROUP BY cid;
7

```

查询结果

▼

Python |

```

1  select * from choiceinfo

```

	cid character (5) 🔒	num bigint 🔒	avg numeric 🔒
1	10001	5898	75.9724011039558418
2	10002	6013	75.8944020356234097
3	10003	5975	75.9430183870380484
4	10004	6110	76.1287462316013478
5	10005	6031	76.0171449196895867
6	10006	6090	76.0699625200785294
7	10007	5965	75.9119038951583546
8	10008	5985	75.6591074020319303
9	10009	5965	76.2119555312556953
10	10010	6027	75.8274111675126904
11	10011	6086	76.3341042519124711
12	10012	5819	76.1280852655198205
13	10013	5982	75.6476068688344903
14	10014	5866	76.1030146106898465
15	10015	5990	75.8496732026143791
16	10016	5867	76.0415129151291513
17	10017	5992	76.0846599131693198

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

▼ Python |

```

1  update students
2  set grade='2002'
3  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

▼ Python |

```

1  update teachers
2  set salary=salary+500
3
```

### 5. 将姓名为zapyv的学生的课程“C”的成绩加上5分

```

1  update choices
2  set score=score+5
3  where sid=
4  (select sid from students where sname='zapyv')
5  and
6  cid=
7  (select cid from courses where cname ='c')
8

```

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

删除前

```

16  select * from students
17  where sname='LiMing'

```

数据输出 消息 通知

	sid [PK] character (9)	sname character varying (30)	email character varying (30)	grade integer
1	700045678	LiMing	LX@cdem	2002

删除后

```

1  delete from students
2  where sname='LiMing'

```

查询结果

```

16 select * from students
17 where sname='LiMing'
18

```

数据输出 消息 通知

SQL

sid [PK] character (9)	sname character varying (30)	email character varying (30)	grade integer
---------------------------	---------------------------------	---------------------------------	------------------

## 7. 删除所有选修课程“Java”的选课记录

```

19 delete from choices
20 where cid=
21 (select cid from courses where cname='java')

```

数据输出 消息 通知

DELETE 6110

耗时445 毫秒 成功返回查询。

## 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

▼

Python |

```

1 delete from courses
2 where hour<48

```

### 会出现报错

错误： 键值对(cid)=(10007)仍然是从表"choices"引用的,在 "courses" 上的更新或删除操作违反了在 "choices" 上的外键约束 "fk\_choices\_courses"

错误： 在 "courses" 上的更新或删除操作违反了在 "choices" 上的外键约束 "fk\_choices\_courses"

SQL 状态：23503

详细： 键值对(cid)=(10007)仍然是从表"choices"引用的。



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

```
23 delete from choices
24 where cid in
25 (select cid from courses where hour<48)
26
27
```

数据输出 消息 通知

DELETE 203571

耗时1 秒 110 毫秒 成功返回查询。

再次执行操作后成功删除

```
27 delete from courses
28 where hour<48
```

数据输出 消息 通知

DELETE 34

耗时787 毫秒 成功返回查询。

# 自我实践

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

Python

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

插入后的查询结果

```
33 select * from students where sname='WangLan'
```

数据输出 消息 通知

SQL

≡+

📄

▼

📋

▼

🗑

🗄

⬇

📈

	sid [PK] character (9)	sname character varying (30)	email character varying (30)	grade integer
1	800022222	WangLan	[null]	[null]

9

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

▼

Python

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

数据输出

消息

通知

INSERT 0 1

耗时70 毫秒 成功返回查询。

插入后的查询结果

```
4 select * from teachers
5 where tname='LXL'
```

数据输出 消息 通知

	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

▼

Python

```
1 update teachers
2 set salary=4000
3 where tid='200010493'
```

4

update teachers

5

set salary=4000

6

where tid='200010493'

数据输出

消息

通知

UPDATE 1

耗时75 毫秒 成功返回查询。

```

1 select * from teachers
2 where tid='200010493'

```

	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

更改前

```

12 select * from teachers
13 where salary<2500

```

数据输出 消息 通知

	tid [PK] character (9)	tname character varying (30)	email character varying (30)	salary integer
1	200005322	udjom	gd5l8@ppmlf.gov	1373
2	200050683	bmwdtty	uw__6@ozpe.edu	1455
3	200066763	btiagykw	dxd09@vxumi.net	1436
4	200120595	pejyw	zu9ee@zrjls.org	1486
5	200132776	rdhznvtcl	kug3q@fyot.net	1421
6	200144197	uvcoff	gni8t7@zag.org	2064
7	200164470	pclrbeuwx	dn5b@czr.org	1459
8	200180180	nibggfks	tr9ej6_@ifmdd.org	2020
9	200207128	utydr	1zt3@tkhna.com	2429
10	200213467	mhlmbix	yau6go@ngt.org	1953
11	200237130	qlgpmj	wnipjcl@wxaf.net	1415
12	200252026	dhighd	dzne@dumic.gov	1396

```

1  update teachers
2  set salary=2500
3  where salary<2500

```

```

8  update teachers
9  set salary=2500
10 where salary<2500

```

数据输出 消息 通知

UPDATE 4014

耗时91 毫秒 成功返回查询。

更改后

```

12 select * from teachers
13 where salary<2500

```

数据输出 消息 通知

SQL

tid	tname	email	salary
[PK] character (9)	character varying (30)	character varying (30)	integer

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

```

1  update choices
2  set tid =
3  (select tid from teachers where tname='rnupx')
4  where tid='200016731'

```

```

15 update choices
16 set tid =
17 (select tid from teachers where tname='rnupx')
18 where tid='200016731'

```

数据输出 消息 通知

UPDATE 7

耗时77 毫秒 成功返回查询。

## 6. 更新编号“800071780”的学生年级为“2001”

Python

```
1 update students
2   grade=2001
3   where sid='800071780'
4
```

1 **update** students  
2 **set** grade=2001  
3 **where** sid='800071780'

数据输出 消息 通知

UPDATE 1

耗时74 毫秒 成功返回查询。

## 7. 删除没有学生选修的课程

存在课程没有学生选择

```
8 select cid from courses where cid not in
9   (select cid
10    from choices
11    group by cid
12    having count(*)>0)
```

数据输出 消息 通知

SQL

cid  
[PK] character (5)

1 10004

删除代码

Python

```
1 delete from courses
2 where cid not in
3   (select cid
4    from choices
5    group by cid
6    having count(*)>0)
```

```
5 delete from courses
6 where cid not in
7 (select cid
8  from choices
9  group by cid
10 having count(*)>0)
```

数据输出 消息 通知

DELETE 1

耗时86 毫秒 成功返回查询。

## 8. 删除年级高于1998的学生信息

```
1 delete from students
2 where
3 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。

错误： 键值对(sid)=(800005753)仍然是从表"choices"引用的.在 "students" 上的更新或删除操作违反了在 "choices" 上的外键约束 "fk\_choices\_students"

错误： 在 "students" 上的更新或删除操作违反了在 "choices" 上的外键约束 "fk\_choices\_students"

SQL 状态： 23503

详细： 键值对(sid)=(800005753)仍然是从表"choices"引用的。

## 更改操作为

```
1 delete from choices
2 where sid in
3 (select sid from students
4  where grade<1998)
```

```
16 delete from choices
17 where sid in
18 (select sid from students
19 where grade<1998)
```

数据输出 消息 通知

DELETE 42247

耗时301 毫秒 成功返回查询。

## 9. 删除没有选修课程的学生信息

```
16 delete from choices
17 where sid in
18 (select sid from students
19 where grade<1998)
```

数据输出 消息 通知

DELETE 46601

耗时56 分 16 秒 成功返回查询。

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

▼ Python |

```
1 delete from choices
2 where score<60
```

数据输出 消息 通知

DELETE 5736

耗时163 毫秒 成功返回查询。

# 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

错误： 键值对(sid)=(800005753)仍然是从表"choices"引用的.在 "students" 上的更新或删除操作违反了在 "choices" 上的外键约束 "fk\_choices\_students"

错误： 在 "students" 上的更新或删除操作违反了在 "choices" 上的外键约束 "fk\_choices\_students"

SQL 状态： 23503

详细： 键值对(sid)=(800005753)仍然是从表"choices"引用的。