

# Database-System Experiment-6

21307289 刘森元

## 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

Macbook Pro 2021 (Apple M1 Pro)

MacOS Ventura 13.5.2

PostgreSQL 15.4 (Homebrew)

Z-shell 5.9

## 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

完整代码见附件 [Experiment-6.sql](#)

使用命令 `\i Experiment-6.sql`，有如下结果

```
BEGIN
ALTER TABLE
DELETE 34
psql:Experiment-6.sql:82: ERROR: insert or update on table "choices" violates
foreign key constraint "fk_choices_courses"
描述: Key (cid)=(10037) is not present in table "courses".
ROLLBACK
INSERT 0 1
INSERT 0 1
UPDATE 1
UPDATE 4014
UPDATE 23
UPDATE 1
DELETE 0
DELETE 40092
DELETE 2
DELETE 21627
```

### 清空数据库并重新导入

由于本次试验涉及到增删，若重复操作初始数据库内容会不一致，所以该项操作是必要的。

```
-- 清空数据库并重新导入
DROP SCHEMA public CASCADE;
CREATE SCHEMA public;
\cd '/Users/qiu_nangong/Documents/Github/Database-System/Experiment-6'
\i STUDENTS.sql
\i TEACHERS.sql
\i COURSES.sql
\i CHOICES.sql
```

### 更改外键约束

表 CHOICES 分别对 STUDENTS(sid), TEACHERS(tid), COURSES(cid) 有外键约束，需要更改为具有级联删除的约束

```
-- 以下操作涉及到删除，修改 CHOICES 中约束为 ON DELETE CASCADE
-- 删除旧的外键约束
ALTER TABLE CHOICES
DROP CONSTRAINT IF EXISTS fk_choices_courses;
ALTER TABLE CHOICES
```

```

DROP CONSTRAINT IF EXISTS fk_choices_students;
ALTER TABLE CHOICES
DROP CONSTRAINT IF EXISTS fk_choices_teachers;
-- 添加新的外键约束
ALTER TABLE CHOICES
ADD CONSTRAINT fk_choices_courses
FOREIGN KEY (cid) REFERENCES COURSES(cid) ON DELETE CASCADE;
ALTER TABLE CHOICES
ADD CONSTRAINT fk_choices_students
FOREIGN KEY (sid) REFERENCES STUDENTS(sid) ON DELETE CASCADE;
ALTER TABLE CHOICES
ADD CONSTRAINT fk_choices_teachers
FOREIGN KEY (tid) REFERENCES TEACHERS(tid) ON DELETE CASCADE;

```

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)

```

-- 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);

```

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)

```

-- 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 cid, COUNT(*) AS num_students, AVG(score) AS average_score
INTO new_table
FROM CHOICES
GROUP BY cid;

INSERT INTO new_table (cid, num_students, average_score)
SELECT cid, COUNT(*) AS num_students, AVG(score) AS average_score
FROM CHOICES
GROUP BY cid;

```

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

```

-- 在STUDENTS表中使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';

```

在TEACHERS表中使用SQL语句将所有教师的工资多加500元

```
-- 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;
```

将姓名为zapyv的学生的课程“C”的成绩加上5分

```
-- 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');
```

在STUDENTS表中使用SQL语句删除姓名为“LiMing”的学生信息

```
-- 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';
```

删除所有选修课程“Java”的选课记录

```
-- 删除所有选修课程“Java”的选课记录
DELETE FROM CHOICES
WHERE cid = (SELECT cid FROM COURSES WHERE cname = 'Java');
```

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

```
-- 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
BEGIN TRANSACTION;
-- 禁用外键约束
ALTER TABLE CHOICES DROP CONSTRAINT FK_CHOICES_COURSES;
-- 删除COURSES表中时间<48的元组
DELETE FROM COURSES
WHERE hour < 48;
-- 启用外键约束
ALTER TABLE CHOICES ADD CONSTRAINT FK_CHOICES_COURSES FOREIGN KEY (cid) REFERENCES
COURSES (cid);
COMMIT;
-- 该删除操作可能受到外键约束的限制，删除操作会失败。在删除之前，需要通过设置外键约束的级联删除或设置NULL来处理相关数据。
```

由上文运行结果

```
psql:Experiment-6.sql:82: ERROR: insert or update on table "choices" violates
foreign key constraint "fk_choices_courses"
描述: Key (cid)=(10037) is not present in table "courses".
```

可知，在关闭约束进行删除后，会出现约束项不匹配的问题。

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

```
-- 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');
```

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

```
-- 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');
```

将TEACHERS表中编号为“200010493”的老师工资改为4000

```
-- 将TEACHERS表中编号为“200010493”的老师工资改为4000
UPDATE TEACHERS
SET salary = 4000
WHERE tid = '200010493';
```

将TEACHERS表中所有工资小于2500的老师工资改为2500

```
-- 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;
```

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

```
-- 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';
```

更新编号“800071780”的学生年级为“2001”

```
-- 更新编号“800071780”的学生年级为“2001”
UPDATE STUDENTS
SET grade = '2001'
WHERE sid = '800071780';
```

删除没有学生选修的课程

```
-- 删除没有学生选修的课程
DELETE FROM COURSES
WHERE cid NOT IN (SELECT DISTINCT cid FROM CHOICES);
```

删除年级高于1998的学生信息

```
-- 删除年级高于1998的学生信息
DELETE FROM STUDENTS
WHERE grade > 1998;
```

删除没有选修课程的学生信息

```
-- 删除没有选修课程的学生信息
DELETE FROM STUDENTS
WHERE sid NOT IN (SELECT DISTINCT sid FROM CHOICES);
```

删除成绩不及格的选课记录

```
-- 删除成绩不及格的选课记录
DELETE FROM CHOICES
WHERE score < 60;
```

## 5. 实验心得

本次试验涉及到使用 PostgreSQL 数据库进行查询操作。通过实践，我对 SQL 查询语句的编写和使用有了更深入的了解。

在实验过程中我学会了：

- 使用 SELECT 语句查询数据库中的数据，并结合 WHERE 子句进行条件筛选。我还了解了如何使用 JOIN 连接多个表，以及如何使用 GROUP BY 子句和聚合函数对数据进行分组和统计。
- 使用子查询来处理复杂的查询需求，例如嵌套查询和使用子查询进行条件筛选。我还了解了如何使用 ORDER BY 子句对查询结果进行排序，并使用 LIMIT 子句限制返回的记录数。

此外，我遇到了一些语法错误和问题，但通过查阅文档和调试，我能够找到并纠正这些错误。这进一步加深了我对 SQL 查询语句的理解和熟练程度。

总的来说，本次试验让我对 PostgreSQL 数据库的查询操作有了更深入的认识，并提升了我的 SQL 查询技能。