东南大学计算机科学与工程学院

数据库原理 实验报告

|  |  |
| --- | --- |
| 姓名 | 牟倪 |
| 学号 | 09019106 |
| 完成日期 | 2021年8月21日 |

# 一、实验要求

* 熟悉MS Access基本操作；
* 建立一个大学数据库，并根据下发的数据确定data schema；
* 将下发的数据导入数据库；
* 生成至少5000条随机学生数据，并将随机学生数据写入数据库。

# 二、实验内容

## 2.1 建立数据库并导入数据

新建.accdb文件university\_A1.accdb，创建表courses、departments、enroll、majors、professors、sections、students，并将对应的.txt文本导入。

各表的data schema如下，其中加粗的为主键：

表1 各数据表的data schema

|  |  |
| --- | --- |
| courses | (**cno**, cname, **dname**) |
| departments | (**dname**, numphds) |
| enroll | (**sid**, grade, **dname**, **cno**, sectno) |
| majors | (**dname**, **sid**) |
| professors | (**pname**, **dname**) |
| sections | (**dname**, **cno**, **sectno**, pname) |
| students | (**sid**, sname, sgender, sage, syear, sgpa) |

以下为确定主键的依据：

* course：cno为课程id，cname为课程名称，dname为开课院系名称。该表记录了课程信息。在表中发现相同id的课程可以由不同院系开设，因此将cno和dname作为主键。
* departments：dname为院系名称，numphds为院系的PhD数量。该表记录了院系信息。院系没有重名的情况，因此将dname作为主键。
* enroll：sid为学生id，grade为学分，dname为开课院系名称，cno为课程id，推测sectno为上课时间。推测该表记录了选课信息，因此sid、cno、dname都应设为主键。
* majors：dname为院系名称，sid为学生id。该表记录了学生所在的专业。每个学生只能在一个专业，因此将sid作为主键。
* professors：pname为教授姓名，dname为教授所在院系。该表记录了教授信息。考虑到教授重名的情况，将pname和dname共同作为主键。
* sections：dname为开课院系名称，cno为课程变化，推测sectno为上课时间，pname为授课教授姓名。推测该表记录了课程与授课时间的对应关系，因此主键应包含课程主键，即cno与dname。同时，在表中发现同一课程可以在不同时间开设，因此将sectno也作为主键。
* students：sid为学生id，sname为学生姓名，sgender为学生性别，sage为学生年龄，syear为学生培养方案的总学年，sgpa为学生成绩。该表记录了学生信息。将sid作为主键。

## 2.2 生成随机学生数据

我们首先观察学生信息数据的特点：

* sid：递增的正整数编号。使用语句“SELECT MAX(sid) FROM students”查询得知，原数据的最大编号位104，因此新数据的编号应从105开始；
* sname：字符串的pattern为“姓氏, 名字首字母.”其中，姓氏的字符串长度大概在4-12，其中第一个字母大写，其他字母都小写。名字首字母为一个大写字母；
* sgender：字符f或字符m；
* sage：15-99之间的正整数；
* syear：1-6之间的正整数；
* sgpa：0-4之间的浮点数，精度为一位小数。

根据数据特点，我们随机生成5000条数据，并将数据写入random\_student.txt文件里。生成随机数据的代码如下（python）：

1. import random
2. import string
3. base, count = 105, 5000
4. with open('random\_student.txt', 'w') as f:
5. upper\_letters = string.ascii\_uppercase
6. lower\_letters = string.ascii\_lowercase
7. for i in range(count):
8. *# generate random student name*
9. rand\_name = '"' + random.choice(upper\_letters)
10. for \_ in range(random.randint(3, 12)):
11. rand\_name += random.choice(lower\_letters)
12. rand\_name += ', ' + random.choice(upper\_letters) + '."'
13. *# generate random record*
14. record = '%d' % (base + i) + '\t' \
15. + rand\_name + '\t' \
16. + ('"m"' if random.randint(0, 2) == 0 else '"d"') + '\t' \
17. + '%d' % random.randint(15, 100) + '\t' \
18. + '%d' % random.randint(1, 6) + '\t' \
19. + '%.2f' % round(random.uniform(0, 4), 1) + '\n'
20. *# print(record)*
21. f.write(record)

将random\_student.txt中的数据导入students表内，我们就得到了5000条随机学生数据。

# 三、实验感悟

我最初的思路是使用SQL视图的查询窗口，直接往access数据库中插入数据。经过测试后发现，查询窗口每次只能执行一条普通语句，难以批量插入数据。并且，仅依靠SQL的逻辑，在插入数据的同时生成随机字符串，也有技术难度。

因此，我尝试用python连接access数据库，利用python的逻辑生成随机字符串，编写for循环代码实现批量插入数据。然而，因为对SQL和access的了解太少，我没有配好环境，最终没能实现python和数据库之间的通信。

后来，受到其他同学的启发，我选择了“先生成数据，再导入外部数据”的方法。该方法简单易行，没有技术难点。

我的感悟是：不要陷入一个思路，要多从各个角度思考。为什么一定要进行数据库操作？为什么不先生成数据再导入呢，就像2.1节中的建表一样？

同时，和同学交流也非常有帮助。在和同学的交流过程中，我还得到了一个只使用access查询窗口的做法：多次执行以下代码，

1. INSERT INTO students
2. SELECT \* FROM students ORDER BY Rnd;

这段代码将原students表内的元素打乱顺序，然后重新插入表格；每执行一次，表格中数据的个数就翻倍。只要执行到数据个数大于5000，我们就得到了5000条数据。虽然数据严格来说不是随机生成的，并且学生的sid字段也需要进行调整（主键不能重复），但这确实是一个巧妙的思路。