数据库原理 第二次实验报告

09019106 牟倪

# 一、实验目的

* 巩固Access基本操作。
* 熟练掌握Access中的SQL查询。

# 二、实验内容

## 2.1 查询1

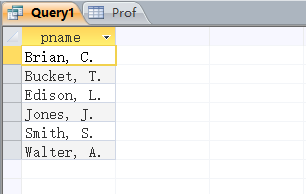
**查询内容**：在{少于50名PhD工作的departments}工作的professor的姓名。

**思路**：关联Dept表和Prof表，从Dept表得到满足条件的department，通过Prof表查询该department下的professor信息。

**SQL语句**：

1. SELECT Prof.pname
2. FROM Prof, Dept
3. WHERE Prof.dname=Dept.dname AND Dept.numphds<50;

**查询结果**：



## 2.2 查询2

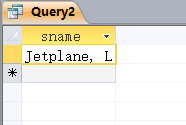
**查询内容**：gpa最低的student的姓名。

**思路**：在Student表查询，找到gpa={最低gpa}的学生，{最低gpa }用嵌套的SELECT子句得到。

**SQL语句**：

1. SELECT sname
2. FROM Student
3. WHERE gpa=(SELECT MIN(gpa) FROM Student);

**查询结果**：



## 2.3 查询3

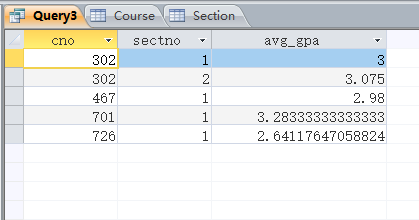
**查询内容**：对每一个Computer Sciences专业的班，查询cno、sectno、学生的平均gpa。

**思路**：关联Enroll表、Student表，从Enroll里筛选得到Computer Sciences专业的选课信息，对cno和sectno进行GROUP BY得到各个班级的组，使用聚合函数查询平均gpa。

**SQL语句**：

1. SELECT Enroll.cno, Enroll.sectno, AVG(Student.gpa) AS avg\_gpa
2. FROM Enroll, Student
3. WHERE Student.sid=Enroll.sid AND Enroll.dname='Computer Sciences'
4. GROUP BY Enroll.cno, Enroll.sectno;

**查询结果**：



## 2.4 查询4

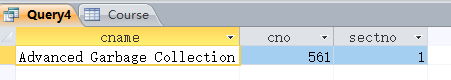
**查询内容**：{所有少于6人参加的course}的cname、cno、sectno。

**思路**：对Enroll进行参数为cno、sectno、dname的GROUP BY操作，把每个班的信息进行聚组，用HAVING筛选出小于6人的班。将Enroll与Course进行关联，得到cname。

**SQL语句**：

1. SELECT Course.cname, Enroll.cno, Enroll.sectno
2. FROM Enroll, Course
3. WHERE Enroll.cno=Course.cno AND Enroll.dname=Course.dname
4. GROUP BY Course.cname, Enroll.cno, Enroll.sectno, Enroll.dname
5. HAVING COUNT(\*)<6;

**查询结果**：



## 2.5 查询5

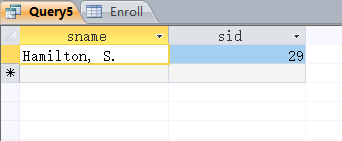
**查询内容**：{参加课程数量最多的student}的sid、sname。

**思路**：首先得到一个子表tempTable，记录了学生的sid、选课数量enrollNum。对enrollNum使用聚合函数MAX，得到最大选课数量。用最大选课数量进行条件查找，得到选课最多的学生们的sid。用该sid去Student里进行索引，得到sid和sname。

**SQL语句**：

1. SELECT Student.sname, Student.sid
2. FROM Student
3. WHERE Student.sid IN(
4. SELECT tsid
5. FROM(
6. SELECT Student.sid AS tsid, COUNT(\*) AS enrollNum
7. FROM Student, Enroll
8. WHERE Student.sid=Enroll.sid
9. GROUP BY Student.sid
10. ) AS tempTable
11. WHERE enrollNUM=(SELECT MAX(enrollNum) FROM (
12. SELECT COUNT(\*) AS enrollNum
13. FROM Student, Enroll
14. WHERE Student.sid=Enroll.sid
15. GROUP BY Student.sid
16. ) AS tempTable)
17. );

**查询结果**：



## 2.6 查询6

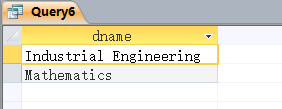
**查询内容**：{拥有小于18岁student}的department的dname。

**思路**：将Major与Student关联，筛选出＜18岁的学生，输出不重复的dname。

**SQL语句**：

1. SELECT DISTINCT Major.dname
2. FROM Major, Student
3. WHERE Major.sid=Student.sid AND Student.age<18;

**查询结果**：



## 2.7 查询7

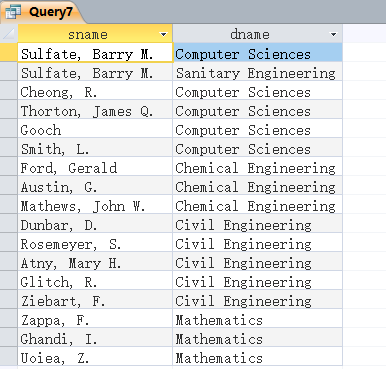
**查询内容**：{参加“College Geometry”相关课程的学生}的sname和dname。

**思路**：用LIKE进行字符串匹配，Access里“\*”代表任意0或多个字符。关联Enroll和Course，筛选符合要求的课程。关联Enroll和Student，得到选课学生的sname。关联Enroll和Major，得到选课学生的专业。

**SQL语句**：

1. SELECT Student.sname, Major.dname
2. FROM Student, Major, Enroll, Course
3. WHERE Student.sid=Major.sid AND Major.sid=Enroll.sid AND
4. Course.cno=Enroll.cno AND Course.dname=Enroll.dname AND
5. Course.cname LIKE '\*College Geometry\*';

**查询结果**：



## 2.8 查询8

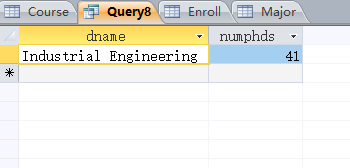
**查询内容**：{没有student参加“College Geometry”相关课程的department}的dname和numphds。

**思路**：首先查询{有student参加“College Geometry”相关课程的dname}，略微修改上一题代码即可得到。查询表Dept，用NOT IN筛选不在{前述搜索结果}中的department。

**SQL语句**：

1. SELECT dname, numphds
2. FROM Dept
3. WHERE dname NOT IN(
4. SELECT Dept.dname
5. FROM Dept, Enroll, Course, Major
6. WHERE Dept.dname=Major.dname AND
7. Enroll.sid=Major.sid AND
8. Enroll.cno=Course.cno AND Enroll.dname=Course.dname AND
9. Course.cname LIKE '\*College Geometry\*'
10. );

**查询结果**：



## 2.9 查询9

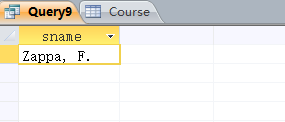
**查询内容**：{同时参与“Computer Sciences”和“Mathematics”专业开设的课程的学生}的sname。

**思路**：查询参与“Computer Sciences”专业开设的课程的学生，并且这个学生还在（IN）{参与“Mathematics”专业开设的课程的学生}的集合里。关联表Student和Enroll，从Enroll筛选dname，从Student查询sname。注意，IN条件嵌套查询中，筛选出的结果是sid，因为sid是主键。

**SQL语句**：

1. SELECT Student.sname
2. FROM Student, Enroll
3. WHERE Student.sid=Enroll.sid AND
4. Enroll.dname='Computer Sciences' AND
5. Student.sid IN(
6. SELECT sid FROM Enroll WHERE dname='Mathematics'
7. );

**查询结果**：



## 2.10 查询10

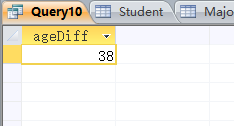
**查询内容**：Computer Sciences专业{最老的学生}与{最年轻的学生}的年龄差值。

**思路**：关联Student、Major得到学生的专业，从Major里筛选出Computer Sciences专业的学生，直接使用聚合函数。

**SQL语句**：

1. SELECT MAX(Student.age)-MIN(Student.age) AS ageDiff
2. FROM Student, Major
3. WHERE Student.sid=Major.sid AND Major.dname='Computer Sciences';

**查询结果**：



## 2.11 查询11

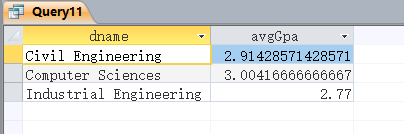
**查询内容**：{存在gpa＜1.0的学生的department}的dname和平均gpa。

**思路**：关联Major、Student，以dname为GROUP BY参数，将各个专业聚成小组。筛出在{存在gpa＜1.0学生的department}集合中的组。{在gpa＜1.0学生的department}集合，直接将Major和Student相关联，筛选出gpa＜1.0的学生，以主键dname为查询的attribute。

**SQL语句**：

1. SELECT Major.dname, AVG(Student.gpa) AS avgGpa
2. FROM Major, Student
3. WHERE Student.sid=Major.sid
4. GROUP BY Major.dname
5. HAVING Major.dname IN(
6. SELECT Major.dname
7. FROM Major, Student
8. WHERE Student.sid=Major.sid AND Student.gpa<1
9. );

**查询结果**：



## 2.12 查询12

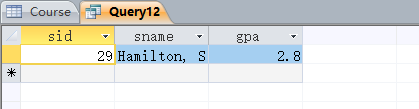
**查询内容**：{同时参加所有“Civil Engineering”学院的课程的学生}的sid、sname、gpa。

**思路**：查找这样的学生：{所有Civil Engineering学院课程的集合 - 学生选的Civil Engineering学院课程的集合}为空集，集合减法用NOT IN，筛选是否空集使用NOT EXISTS。即，没有一门Civil Engineering学院课程是没被选的。

**SQL语句**：

1. SELECT Student.sid, Student.sname, Student.gpa
2. FROM Student
3. WHERE NOT EXISTS(
4. SELECT cno FROM Course WHERE dname='Civil Engineering' AND
5. cno NOT IN
6. (SELECT Enroll.cno FROM Enroll WHERE Enroll.sid=Student.sid AND Enroll.dname='Civil Engineering')
7. );

**查询结果**：



# 三、实验总结

经过12个题的练习，我对解决SQL查询问题的逻辑有了更多认识，SQL基本查询操作也更加熟练。Access SQL的语法和课程内容不尽相同，如’\*’而非’%’表示匹配0或多个字符，类似的经验需要在实践中积累。