1. **实验目的：**
2. **了解SQL语言的特点和基本概念。**
3. **能够针对某种具体的DBMS（本实验采用Access2016），熟练地运用单表查询、连接查询、嵌套查询、集合查询等各种SQL查询语句对数据库中的表进行操作。**
4. **对相同的查询要求，能够采用多种查询方法实现，并能分析各种方法的优劣， 从中选择合适的方法。**
5. **实验过程：**

实验一**：**Print the names of professors who work in departments that have fewer than 50 PhD students.

解：（1）分析：本题是查询在博士生人数少于50个人的系工作的教师名字。查询教授名字可以通过prof表，而所查询的教授名字是有限制条件的，

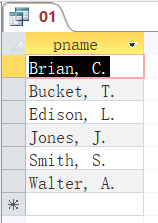
他所工作的系名要属于特定的集合（所有系名的一个子集），显然这个子集可以通过查询dept表获得，所以带有谓词in的嵌套子查询可以实现题目要求。

（2）语句实现：

SELECT pname FROM Prof

WHERE dname IN (SELECT dname FROM Dept WHERE numphds < 50);

（3）查询结果：



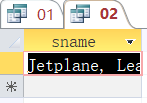
实验二：Print the name(s) of student(s) with the lowest gpa

解：（1）分析：本题是查询成绩点最低的学生的名字。最低的成绩点可以在student表中通过函数min(gpa)获得，而所查询学生的名字的限制为成绩点等于min(gpa), 因此可用如下嵌套子查询实现。

（2）语句实现：SELECT sname FROM student

WHERE gpa IN (SELECT min(gpa) FROM student);

（3）查询结果：



实验三：For each Computer Sciences class, print the cno, sectno, and the average gpa of the student enrolled in the class.

解：（1）分析：本题是查询计算机科学系的所有班的课程号、分班号、班上学生的平均绩点。计算机科学系的所有班可以通过section表获得, 而通过enroll表可以由section表中的dname, cno, sectno获得班上所有学生的sid，而通过sid可以在student表中查得学生成绩点，最后由cno, sectno进行分组，并用函数avg(gpa)，获得每组的平均成绩。所以可用三个表的连接查询，并适当分组实现查询要求。

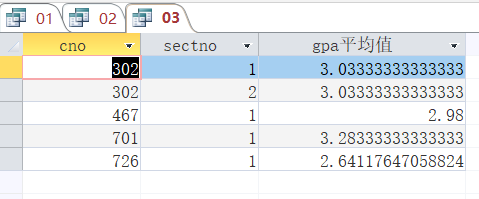
（2）语句实现：

SELECT section.cno, section.sectno, Avg(gpa) AS gpa之平均值

FROM [section], enroll, student

WHERE ( (section.dname ='Computer Sciences') and (section.cno = enroll.cno) and (enroll.sid=student.sid) )

GROUP BY section.cno, section.sectno; （3）查询结果：



实验四：Print the course names, course numbers and section numbers of all classes with less than six students enrolled in them.

解：（1）分析：本题是查询所有班级人数少于6的课程名，课程号，分班号。通过section表可以查询出所有的班，其中的课程名可由查询所得的dname, cno在course表中确定，因为与班级人数有关，还需将section表和enroll表做连接，并按section.cno, section.dname, section.sectno分组以获取所有班的人数。所以可用连接查询、嵌套查询，并适当分组来实现查询要求。

（2）语句实现：

SELECT (select cname from course where cno=section.cno and dname=section.dname) AS cname, section.cno, section.sectno

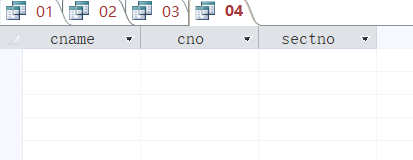
FROM [section], enroll

WHERE (section.cno = enroll.cno )

GROUP BY section.cno, section.sectno, section.dname

HAVING count(\*) < 6;

（3）查询结果：没有低于6个的



实验五：Print the name(s) and sid(s) of student(s) enrolled in the most classes.

解：（1）分析：本题是查询加入班级数最多的学生名字、学号。可采用临时表的方法解决。建立一个临时表包括字段学生名字、学号和所选课程的数目（num）。再对这个临时表进行查询（名字、学号），限制条件是所选课程数目等于max(num)。

（2）语句实现：

SELECT sid, sname

FROM (SELECT sid, (select sname from student where sid=enroll.sid) AS sname , count(\*) As num

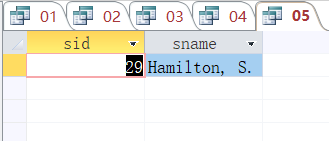
FROM enroll

group by enroll.sid) AS b

WHERE b.num = ( select max(num) From (SELECT sid, (select sname from student where sid=enroll.sid) AS sname , count(\*) As num

FROM enroll group by enroll.sid) );

(3)查询结果：



实验六：Print the names of departments that have one or more majors who are under 18 years old.

解：（1）分析： 本题是查询所含学生至少有一个年龄小于18岁的系的名称。在major

表中可以得到每个学生所属专业情况，该系存在学生年龄小于18岁的学生（通过student表）则该系满足要求。因此可用带有exists的嵌套子查询实现要求。

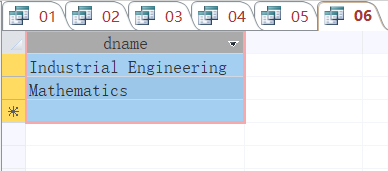
（2）语句实现：SELECT dname

FROM major

WHERE exists(

select \* from student

where student.age < 18 and student.sid = major.sid ); （3）查询结果：



实验七：Print the names and majors of students who are taking one of the College Geometry courses.

解：（1）分析：本题查询所有选了College Geometry courses的学生的姓名和所在系名。此题对课程名需要用通配符进行模糊查询。学生的限制条件是存在一门College Geometry course为他所选。因此还需用到带有exists的嵌套子查询。

（2）语句实现：

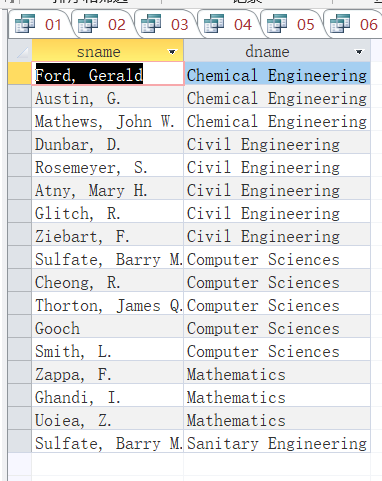
SELECT (select sname from student where sid=major.sid) AS sname, dname

FROM major

WHERE exists(

select \* from course,enroll where enroll.dname = 'Mathematics' and enroll.cno in (461,462) and enroll.sid = major.sid);

（3）查询结果：



实验八：For those departments that have no majors taking a College Geometry course, print the department name and the number of PhD students in the department.

解：（1）分析：本题是查询所含学生都没有选College Geometry course的系的名称和该系的博士生人数。本题思路和方法与上题基本一致。

（2）语句实现：

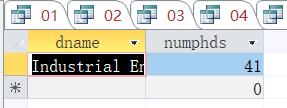
SELECT dname, numphds

FROM dept

WHERE not exists(

select \* from enroll,major where enroll.dname = 'Mathematics' and enroll.cno in (461,462) and enroll.sid = major.sid and major.dname = dept.dname);

（3）查询结果：



实验九：Print the names of students who are taking both a Computer Sciences course and a Mathematics course.

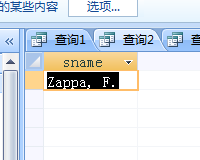
解：（1）分析： 本题是查询既选了计算机科学系课程又选了数学系课程的学生姓名是求两个集合交的问题。因为一般的DBMS没有实现交操作，可以先根据一个条件获取一个集合然后根据第二个条件筛选，即第一个集合要选取也在第二个条件产生的集合元素中。因此可以用带有in的嵌套子查询完成要求。

（2）语句实现：

SELECT sname

FROM student

WHERE sid in (select sid from enroll where dname="Computer Sciences") and sid in(select sid from enroll where dname="Mathematics"); （3）查询结果



实验十：Print the age difference between the oldest and youngest Computer Sciences major(s).

解：（1）分析：本题是查询计算机科学性年龄最大的学生和年龄最小的学生之间的年龄差。思想还是通过先建立一张临时表，包括字段系名、学号、年龄。再在这张临时表里利用max(age)， min(age)获取最大年龄和最小年龄，进行相减运算。

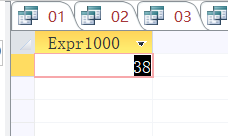
（2）语句实现：

SELECT Max(age)-Min(age) AS 表达式1

FROM student

WHERE (((student.[sid]) In (select sid from major where dname="Computer Sciences")));4

（3）查询结果：



实验十一：For each department that has one or more majors with a GPA under 1.0, print the name of the department and the average GPA of its majors.

解：（1）分析：本题是查询系里有学生绩点小于1.0的系的系名和该系学生的平均绩点。本题主要是对major， student两张表按sid做连接，连接后再按dname分组，但要注意将min(gpa)<1的分组剔除。

（2）语句实现：

SELECT dname, avg(gpa) AS avgpa

FROM major, student

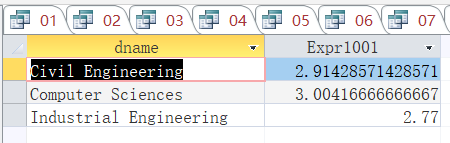
WHERE dname in(SELECT dname

FROM major, student

where major.sid = student.sid and student.gpa < 1.0

group by dname) and major.sid = student.sid

GROUP BY dname; （3）查询结果：



1. 实验总结
2. 通过本次实验基本了解SQL查询语句的运用，但是并不是很熟练。
3. 数据导入时碰到了一些问题，默认数字形式不保留小数位。
4. 掌握了SQL语句在特定情况下的一些运用技巧：
   1. 对有多条件约束的较难的查询操作可考虑使用集合的交、并等操作思想。一般的DBMS都直接实现了并操作，交操作可用其它间接方法实现。如第9题。
   2. 对有层次关系的查询，即子查询的结果可以作为父查询的查找条件时，可考虑使用嵌套查询。如1、2题。
   3. 对涉及到多表之间关系的查询，可以考虑使用表的连接操作，但要理清表与表之间的连接条件。如4、11题。