

实验五 SQL server 数据查询操作

【实验目的】

1. 掌握 SELECT 命令的高级用法。
2. 能使用 where 子句进行各种条件的查询和多表的联接,其中包括子句 IN、NOT IN 和 BETWEEN, 以及 LIKE、ORDER BY 子句, Exist 和 not Exist, All 和 Any 等的用法。
3. 掌握聚合函数的用法。
4. 掌握 GROUP BY 与 HAVING 子句的用法。

【实验学时】

建议 2 学时

【实验环境配置】

1. SQL SERVER 环境

【实验原理】

SQL select 语言

【实验步骤】

1、列出选修了所有课程的学生名单

方案一:

```
USE university_db
GO

SELECT sname 姓名
FROM student
where NOT EXISTS
(SELECT *
FROM course
where NOT EXISTS
(SELECT *
FROM sections
where EXISTS
(SELECT *
FROM sc
where sno=student.sno and secno=sections.secno
and course.cno=sections.cno
)
)
)
```

方案二:

```
SELECT distinct sname as 姓名
FROM student,sc,sections
where student.sno=sc.sno and
sc.secno=sections.secno
```

```
GROUP BY sname
HAVING COUNT(sections.cno)=
(select COUNT(*) from course)
```

--先连接 student, sc, sections 表, 按照 sname 分组

--来求一个学生选课的总数再与 count(course) 课程总数比较

--'=' 即选修了所有的表

执行结果:



The screenshot shows a database query result window. At the top, there is a SQL statement: `(SELECT * FROM`. Below the statement, there are two tabs: '结果' (Results) and '消息' (Messages). The '结果' tab is active, displaying a table with two columns: '姓名' (Name) and an index column. The table contains one row with the name '赵剑' (Zhao Jian) and the index '1'.

	姓名
1	赵剑

2、检索每门课程成绩都在 80 分以上的学生名单

方案一:

```
SELECT sname 姓名
FROM student
where sno in
(select grade from sc b
where grade <= 80 and
a.sno = b.sno
)
```

选择两个 sc 表 a、b, 遍历表中的 grade,

若发现 $grade \leq 80$ 且 a 表选中的 sno 和 b 表中的 sno 匹配, 此时 not exists 语句中为假, 外层选中的 sno 不能输出, 即过滤掉了成绩 $grade \leq 80$ 的学生 sno。如果外层的 sno 进入 not exists 语句后查找 b 表找不到成绩 ≤ 80 的记录, 此时语句为空值, 可以输出, 即该生的每门课成绩都 > 80

方案二:

```
select sname 姓名 from student,sc
where student.sno=sc.sno
group by(sname)
having MIN(grade)>80
```

--连接学生表 student 和 sc 学生课程成绩表, 以 sname 为分组, 则每组的记录读书有关一个学生的成绩信息, 取最小值 > 80也即每门课都在 80 分以上。

执行结果:

结果

消息

姓名

1

王谦

2

孙启明

3

宇帆

4

海琳

姓名

1

海琳

2

孙启明

3

王谦

4

宇帆

查询已成功执行。

3、检索获奖学金的学生名单（每门课程在 80 分以上，平均成绩在 90 分以上）

方案一:

```
select sname as 姓名
from student,sc
where student.sno=sc.sno
group by sname,student.sno
having MIN(grade)>80 and AVG(grade)>90
```

--如上题,该题只是在基础上加了一个平均分大于 90 的条件，也是将叙述 sno 和成绩 grade 关联起来，再用 AVG 函数判断平均成绩。

方案二:

```
select sname 姓名 from student
where sno in(
    select sno from sc a
    where not exists
    ( select grade from sc b
      where grade<80 and a.sno=b.sno
      group by sno
    )
    having AVG(grade)>90
)
```

--这里先选出那些每门课成绩都在 80 分以上的同学，再用 sno 分组找出平均分在 90 分以上的同学。

执行结果:



	姓名
1	王谦

	姓名
1	王谦

4、查询选修了数据库技术、没有选修高等数学的学生姓名和系别

方案一:

```
select sname 姓名, sdept 系别
from
student, sc, sections, course
where student.sno=sc.sno and
sc.secno=sections.secno and
sections.cno=course.cno
and cname LIKE '数据库技术' and
sname not in (
select sname from
student, sc, sections, course
where student.sno=sc.sno and
sc.secno=sections.secno and
sections.cno=course.cno and
cname LIKE '高等数学'
)
```

因为本题没有用到有关成绩的信息，所以可以先不用 sc 表，再分别来凝结四个表，并根据 cname 条件找出题解

方案二:

```
select sname 姓名, sdept 系别
from student
where EXISTS
(select * from
sc, sections, course
where student.sno=sc.sno and
sc.secno=sections.secno and
sections.cno=course.cno and
cname LIKE '数据库技术'
)
AND NOT EXISTS(
select * from
sc, sections, course
where student.sno=sc.sno and
sc.secno=sections.secno and
sections.cno=course.cno and
cname LIKE '高等数学'
)
```

-- 逐次从 student 表中取出一行，与其他三个表建立连接，根据 EXISTS 和 NOT EXISTS 语句是否为空输出(同时满足两个条件)。

执行结果:



	姓名	系别
1	宇帆	机械工程

5、统计所有课程的最高成绩、最低成绩和平均成绩

方案一:

```
select course.cname 课程名 ,MAX(grade) 最高分,MIN(grade) 最低分,  
AVG(grade) 平均分  
from student,course,sections,sc  
where student.sno=sc.sno and  
sc.secno=sections.secno and  
course.cno=sections.cno  
group by course.cno,cname
```

--因为要查询课程、分数，所以四个表都要连接，因为查询的是每门课的平均分，所以根据课程名称和编号分组(任选一)，每组里面都是关于一个课程的，用 AVG 求平均分。

执行结果:

<				
结果 消息				
	课程名	最高分	最低分	平均分
1	大学英语	90	56	83
2	高等数学	94	80	88
3	大学物理	93	78	87
4	数据库技术	90	90	90
5	多媒体技术	98	92	94
6	VB程序设计	94	50	77

表 1 university1 数据库的结构定义

表名	属性名	描述	类型	约束
student	sno	学号	字符类型，长度为 6	主键，不可为空 *首字母必须为 s
	sname	姓名	字符类型，长度为 10	不可为空
	ssex	性别	字符类型，长度为 2	不可为空 *只能取值‘男’或‘女’
	sbirth	出生日期	日期类型	不可为空
	sdept	系部	字符类型，长度为 20	不可为空
course	cno	课程号	字符类型，长度为 6	主键，不可为空 *首字母必须为 c
	cname	课程名	字符类型，长度为 20	不可为空
	cpno	先导课	字符类型，长度为 6	外键（参照 course 表 cno）
	ccredit	学分	整数类型	不可为空 *取值介于 0~8 之间
sections	secno	班级号	字符类型，长度为 8	主键，不可为空
	cno	课程号	字符类型，长度为 6	不可为空， 外键（参照 course 表 cno）
	pno	教师号	字符类型，长度为 6	*首字母必须为 p
sc	sno	学号	字符类型，长度为 6	不可为空，主属性， 外键（参照 student 表 sno）
	secno	班级号	字符类型，长度为 8	不可为空，主属性， 外键（参照 sections 表 secno）
	grade	成绩	整数类型	不可为空 *取值介于 0~100 之间

表 2 student 表数据

	sno	sname	ssex	sbirth	sdept
1	s001	赵剑	男	1987-03-25 00:00:00.000	计算机
2	s002	王谦	男	1986-01-01 00:00:00.000	交通工程
3	s003	孙启明	男	1987-04-01 00:00:00.000	土木工程
4	s004	宇帆	男	1987-09-17 00:00:00.000	机械工程
5	s005	李晓静	女	1988-06-21 00:00:00.000	生物工程
6	s006	金之林	女	1988-06-21 00:00:00.000	计算机
7	s007	张东晓	男	1987-08-02 00:00:00.000	城市规划
8	s008	海琳	女	1988-07-23 00:00:00.000	城市规划

表 3 course 表数据

	cno	cname	cpno	ccredit
1	c116	大学英语		6
2	c120	高等数学		6
3	c126	大学物理	c120	3
4	c130	数据库技术		3
5	c132	多媒体技术	c120	3
6	c135	VB程序设计	c130	3

表 5 sc 表数据

	sno	secno	grade
1	s001	11601	77
2	s001	12001	80
3	s001	12601	89
4	s001	13002	90
5	s001	13201	92
6	s001	13501	94
7	s002	11602	90
8	s002	12601	88
9	s002	13201	98
10	s003	11601	90
11	s003	12002	94
12	s003	12601	88
13	s004	11601	89
14	s004	13001	90
15	s004	13201	92
16	s004	13501	89
17	s005	11602	56
18	s006	11601	88
19	s006	12601	78
20	s007	11602	90
21	s007	13201	95
22	s007	13501	50
23	s008	11601	89
24	s008	12001	90
25	s008	12601	93

表 4 sections 表数据

	secno	cno	pno
1	11601	c116	p001
2	11602	c116	p002
3	12000	c135	p007
4	12001	c120	p003
5	12002	c120	p003
6	12601	c126	p004
7	13001	c130	p005
8	13002	c130	p006
9	13201	c132	p007
10	13501	c135	p007