

一. 实验目的

1. 熟悉SQL的数据查询语言
2. 能够使用SQL语句对数据库进行嵌套查询、集合运算。

二. 实验内容

1. 嵌套查询

- 通过实验验证对子查询的两个限制条件：不能使用order by子句；外层select语句变量可以用在子查询中，但反之不行。
- 体会相关子查询和不相关子查询的不同：相关子查询要使用外层select语句提供的数据；不相关子查询即内层子查询不依赖于外层select语句。
- 考察4类谓词的用法,包括:
 - 第1类,IN, NOT IN
 - 第2类,带有比较运算符的子查询
 - 第3类,SOME,ANY或ALL谓词的子查询
 - 第4类,带有 EXISTS谓词的子查询。

2. 连接查询

- 使用保留字 UNION、INTERSECT、EXCEPT进行集合或、交、减运算。

三. 实验结果

1. 查询选修C++课程的成绩比姓名为 ZNKOO的学生高的所有学生的编号和姓名

先找出ZNKOO的c++成绩，然后在找出c++成绩比ZNKOO高的学生

```
SELECT DISTINCT STUDENTS.sid as '学号',  
STUDENTS.sname as '学生姓名'  
FROM CHOICES,STUDENTS,COURSES  
WHERE CHOICES.cid=COURSES.cid  
AND COURSES.cname='c++'  
AND CHOICES.sid=STUDENTS.sid  
AND CHOICES.score>(  
SELECT CHOICES.score  
FROM CHOICES,STUDENTS,COURSES  
WHERE STUDENTS.sname='znkoo'  
AND CHOICES.sid=STUDENTS.sid  
AND COURSES.cname='c++'  
AND CHOICES.cid=COURSES.cid  
)
```

结果如下：

	学号	学生姓名
1	807480998	mikcbr
2	808000007	sanhbgh
3	808020819	mcjdrtoe
4	823607770	hjaoqtj
5	823891641	iafzixl
6	824201902	utyhpoyu
7	824682545	zvsgacq
8	824822641	izimp
9	824985973	lpyikli
10	825130104	zmyjtus
11	838411799	yzcfwic
12	839282746	seoksdre
13	839500666	faihrdssd

查询已成功执行。 | (local)\CHONOR (10.0 RTM) | DESKTOP-CHONOR\Chonor ... | School | 00:00:00 | 2048 行

2. 找出和学生883794999或学生850955252的年级一样的学生的姓名
先找出学生883794999，学生850955252的年级，然后再找出年级一样的学生

```
SELECT sname as '学生姓名',
grade as '年级'
FROM STUDENTS
WHERE grade IN(
    SELECT grade
    FROM STUDENTS
    WHERE sid='883794999' OR sid='850955252'
)
```

结果如下：

	学生姓名	年级
1	uxqgbkjin	1997
2	hvyfafcj	1997
3	vocojjinml	1997
4	ztozk	2001
5	efmgfprz	1997
6	pgrimkdhh	2001
7	ttyyb	1997
8	mcefir	1997
9	rmbgu	1997
10	nkdnfq	2001
11	gmxesrhv	1997
12	fbbiyeh	1997

查询已成功执行。 | (local)\CHONOR (10.0 RTM) | DESKTOP-CHONOR\Chonor ... | School | 00:00:00 | 13372 行

3. 查询没有选修Java的学生名称
先找出选修Java的学生名字然后用NOT IN排除

```

SELECT sname as '学生姓名'
FROM STUDENTS
WHERE sname NOT IN(
    SELECT STUDENTS.sname
    FROM COURSES,STUDENTS,CHOICES
    WHERE COURSES.cname='java'
    AND CHOICES.cid=COURSES.cid
    AND STUDENTS.sid=CHOICES.sid
)

```

结果如下：

学生姓名
1 waqcj
2 uxqqbkjn
3 zapyv
4 cxjnj
5 tkbzqduq
6 mglhxm
7 yahvv
8 zvjteiv
9 vfkmnuw
10 wnsphiv
11 nkdnfq
12 tlqghcta

查询已成功执行。 | (local)\CHONOR (10.0 RTM) | DESKTOP-CHONOR\Chonor ... | School | 00:00:00 | 94046 行

4. 找出课时最少的课程的详细信息

```

SELECT * FROM COURSES WHERE COURSES.hour<=ALL(
    SELECT hour FROM COURSES WHERE hour IS NOT NULL
)

```

结果如下：

cid	cname	hour
1 10024	use case	18
2 10034	windows	18

查询已成功执行。 | (local)\CHONOR (10.0 RTM) | DESKTOP-CHONOR\Chonor ... | School | 00:00:00 | 2 行

5. 查询工资最高的教师的编号和开设的课程号

先找出工资最高的教师，他们在找出开课编号

```

SELECT TEACHERS.tid as '教师编号',
CHOICES.cid as '课程编号'
FROM CHOICES,TEACHERS
WHERE TEACHERS.tid=CHOICES.tid
AND TEACHERS.salary>=ALL(
    SELECT salary
    FROM TEACHERS
    WHERE salary IS NOT NULL
)

```

结果如下：

	教师编号	课程编号
1	277877392	10041
2	204711560	10001
3	204711560	10026
4	287866460	10005
5	277877392	10047
6	214445507	10012
7	287866460	10004
8	214445507	10005
9	204711560	10003
10	287866460	10019
11	204711560	10006
12	277877392	10004

查询已成功执行。 | (local)\CHONOR (10.0 RTM) | DESKTOP-CHONOR\Chonor ... | School | 00:00:00 | 84 行

6. 找出选修课程ERP成绩最高的学生编号

先找出ERP课程的最高分，在找出学生编号

```

SELECT STUDENTS.sid as '学号',
STUDENTS.sname as '学生姓名'
FROM CHOICES,STUDENTS,COURSES
WHERE CHOICES.sid=STUDENTS.sid
AND CHOICES.cid=COURSES.cid
AND COURSES.cname='erp'
AND CHOICES.score>=ALL(
    SELECT CHOICES.score
    FROM CHOICES,COURSES
    WHERE COURSES.cname='erp'
    AND CHOICES.cid=COURSES.cid
    AND CHOICES.score IS NOT NULL
)

```

结果如下：

结果		消息
学号	学生姓名	
1	866949513	spxznwpf
2	804483690	phmtksan
3	844447081	pnugb
4	826003101	imwjpmvx
5	816867292	bjzcmsnt
6	853145416	vackjlzsj
7	867662843	ctgsc
8	803973094	xrtike
9	823718255	jubxco
10	894543567	kzaqu
11	871367805	vdqazcy
12	862976650	xuucz

查询已成功执行。 (local)\CHONOR (10.0 RTM) DESKTOP-CHONOR\Chonor ... School 00:00:00 95 行

7. 查询没有学生选修的课程名称

找出CHOICES 所有的课程编号，在COURSE中排除

```
SELECT COURSES.cname as '课程名称'
FROM COURSES
WHERE COURSES.cid NOT IN(
    SELECT DISTINCT cid
    FROM CHOICES
)
```

结果如下：

结果		消息
课程名称		

查询已成功执行。 (local)\CHONOR (10.0 RTM) DESKTOP-CHONOR\Chonor ... School 00:00:00 0 行

8. 查询讲授课程UML的教师所讲授的所有课程名称

先找出教授UML的教师，在找出其开设课程的cid，再去找出cname

```

SELECT DISTINCT cname as '课程名称'
FROM COURSES
WHERE cid IN(
    SELECT DISTINCT cid
    FROM CHOICES
    WHERE tid IN (
        SELECT DISTINCT CHOICES.tid
        FROM CHOICES, COURSES
        WHERE CHOICES.cid=COURSES.cid
        AND COURSES.cname='uml'
    )
)

```

结果如下:

课程名称	
1 algorithm	
2 architectonics	
3 artfical intelligence	
4 asp	
5 basic	
6 c	
7 c#	
8 c++	
9 c++/stl programming	
10 cobol	
11 compiling principle	
12 computer graphics	

查询已成功执行。 | (local)\CHONOR (10.0 RTM) | DESKTOP-CHONOR\Chonor ... | School | 00:00:00 | 50 行

9. 使用集合交运算，查询既选修了database又选修了UML课程的学生编号

```

SELECT sid as '学号'
FROM CHOICES
WHERE cid IN(
    SELECT cid FROM COURSES WHERE COURSES.cname='database'
)
INTERSECT
SELECT sid
FROM CHOICES
WHERE cid IN(
    SELECT cid FROM COURSES WHERE COURSES.cname='uml'
)

```

结果如下:

结果		消息
	sid	
1	801619231	
2	806238872	
3	820098772	
4	821126308	
5	824825346	
6	828934699	
7	830644918	
8	834921972	
9	836670359	
10	838450830	
11	838803879	

查询已成功执行。 (local)\CHONOR (10.0 RTM) DESKTOP-CHONOR\Chonor ... School 00:00:00 303 行

10. 使用集合减运算，查询选修了database却没有选修UML课程的学生编号

```

SELECT sid as '学号'
FROM CHOICES
WHERE cid IN(
    SELECT cid FROM COURSES WHERE COURSES.cname='database'
)
EXCEPT
SELECT sid
FROM CHOICES
WHERE cid IN(
    SELECT cid FROM COURSES WHERE COURSES.cname='uml'
)

```

结果如下：

结果		消息
	学号	
1	800230908	
2	800266130	
3	800654557	
4	800898608	
5	800988010	
6	801147943	
7	801188913	
8	801230082	
9	801266324	
10	801332759	
11	801597743	

查询已成功执行。 (local)\CHONOR (10.0 RTM) DESKTOP-CHONOR\Chonor ... School 00:00:00 5454 行

四. 实验感想

这次实验是学习嵌套查询和集合运算，实际使用时嵌套查询和集合运算大多数可以相互转换，集合运算在实际使用中思路比嵌套查询更加简单，但是查询复杂度来说集合运算复杂度更高效率更低，嵌套查询使用中需要有一定的逻辑思维需要倒着想才能先打外层代码，不然就把代码先打内层查询，然后在转到外层查询。