中山大學
SUN YAT-SEN UNIVERSITY

本科生实验报告

学生姓名：　丁晓琪

学生学号：　22336057

专业名称：　计科

一：实验要求

(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;

(2)找出和学生883794999或学生850955252的年级一样的学生的姓名;

(3)查询没有选修Java的学生名称

(4)找出课时最少的课程的详细信息;

(5)查询工资最高的教师的编号和开设的课程号

(6)找出选修课程ERP成绩最高的学生编号;
(7)查询没有学生选修的课程名称;
(8)查询讲授课程UML的教师所讲授的所有课程名称;
(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;
(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;

二：实验过程

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：

```
use School
select STUDENTS.sid, sname
from STUDENTS, CHOICES, COURSES
where STUDENTS.sid=CHOICES.sid and COURSES.cname='c++' and CHOICES.cid=COURSES.cid and CHOICES.score>(select score
                                                                                                      from CHOICES, STUDENTS, COURSES
                                                                                                      where STUDENTS.sname='ZNKOO' and STUDENTS.sid=CHOICES.sid and COURSES.cid=CHOICES.cid and COURSES.cname='c++')
```

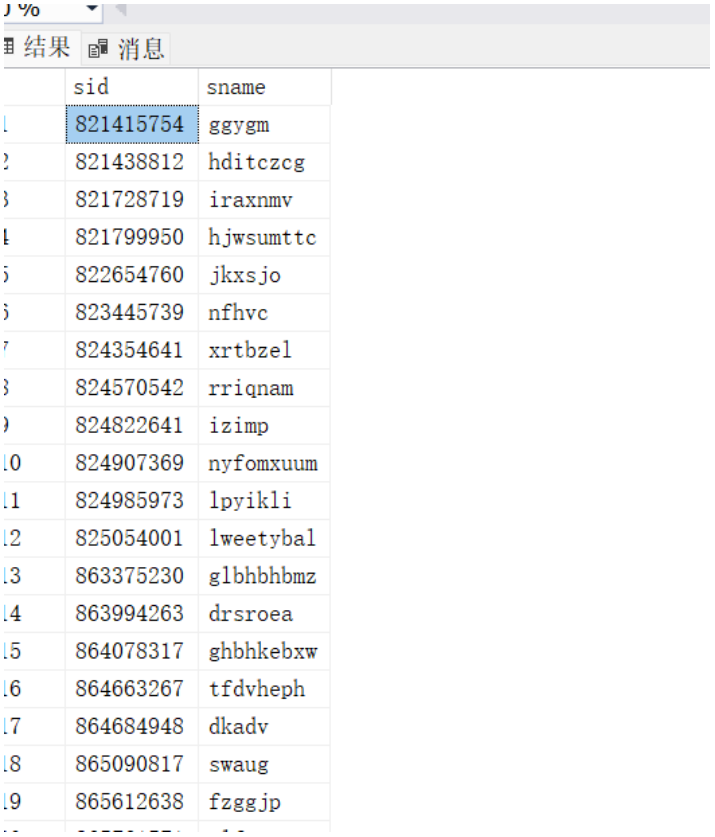
修改版：要加上`distinct`,可能有人修过两次C++

```
use School
select distinct STUDENTS.sid, sname
from STUDENTS, CHOICES, COURSES
where STUDENTS.sid=CHOICES.sid and COURSES.cname='c++' and CHOICES.cid=COURSES.cid and CHOICES.score>(select score
                                                                                                      from CHOICES, STUDENTS, COURSES
                                                                                                      where STUDENTS.sname='ZNKOO' and STUDENTS.sid=CHOICES.sid and COURSES.cid=CHOICES.cid and COURSES.cname='c++')
```

   - `from`子句解析：

     1. 和3. 将`students`和`choices`和`courses`拼接起来，将每个学生和该学生的选课信息还有选择课程的课程具体信息拼接
     2. 筛选课程为'C++'的选课记录
     3. 先用子查询查询出名字为'ZNKOO'的C++课程的分数，再用比较符限定父查询要筛选出分数大于'ZNKOO'的C++课程的分数的元组

- 结果：



	sid	sname
1	821415754	ggygm
2	821438812	hditczcg
3	821728719	iraxnmv
4	821799950	hjwsumttc
5	822654760	jkxsjo
6	823445739	nfhvc
7	824354641	xrtbzel
8	824570542	rriqnam
9	824822641	izimp
10	824907369	nyfomxuum
11	824985973	lpyikli
12	825054001	lweetybal
13	863375230	glbhbhbmz
14	863994263	drsroea
15	864078317	ghbhkebxw
16	864663267	tfdvheph
17	864684948	dkadv
18	865090817	swaug
19	865612638	fzggjp

2. 找出和学生883794999或学生850955252的年级一样的学生的姓名：
   - `where`子句子查询：在关系`students`查询出学号为883794999和850955252的年级（用逻辑符`or`筛选元组）
   - `in`：在关系`students`中选出年级属性取值在子查询的结果关系中的元组
   -
```
use School
select sname
from STUDENTS
where STUDENTS.grade in (select grade
                         from STUDENTS
                         where STUDENTS.sid='883794999' or STUDENTS.sid='850955252')
```
   - 结果：

结果 消息

	sname
1	uxqqbkjn
2	hvylafcj
3	vocojjnml
4	ztozk
5	efmgfprz
6	pgmrkdhh
7	ttyyb
8	mcefjr
9	rlmbgu
10	nkdnfq
11	grnxesrhv
12	fbbiyeh
13	qtxblqzsv
14	ctvxn
15	airnnfv
16	vaesalave
17	uiuirl
18	bqfsghv
19	ieiojxooe

3. 查询没有选修java的学生名称：

```
use School
select sname
from STUDENTS
EXCEPT
select sname
from STUDENTS,CHOICES
where STUDENTS.sid=CHOICES.sid and CHOICES.cid=(select cid
                                                from COURSES
                                                where cname='java')
```

1. 选出所有学生

2.集合差

3.选出选修过java课程的学生

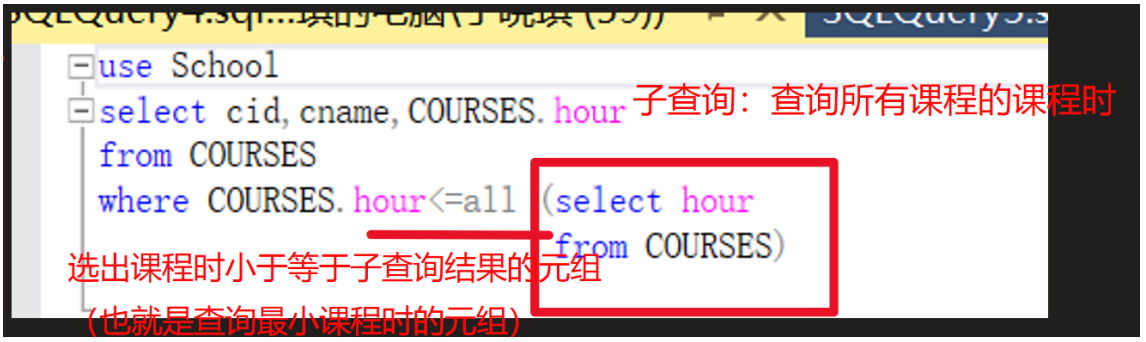
子查询：查询java的cid

结果：

结果 消息

	sname
1	kgqtyoae
2	eeevmdubl
3	tfvhiiq
4	rmzpu
5	ykvdp
6	mqmxagl
7	bodvqfaxj
8	ksewonl
9	isyajaeh
10	ojezog
11	ipyyxsuy
12	dctuaqgu
13	vfpribkof
14	ipmcap
15	bjngiiqzd
16	afvvnyhk
17	iqnaf
18	qscyaygx
19	ecjhkcvsx

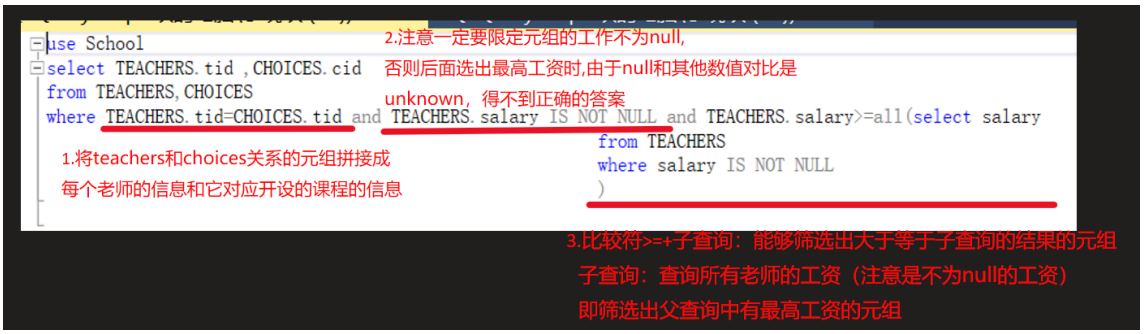
4. 找出课时最少的课程的详细信息



结果：

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

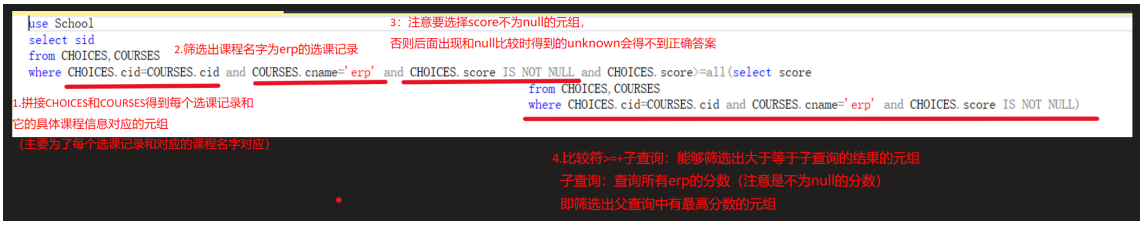
5. 查询工资最高的教师的编号和开设的课程号：



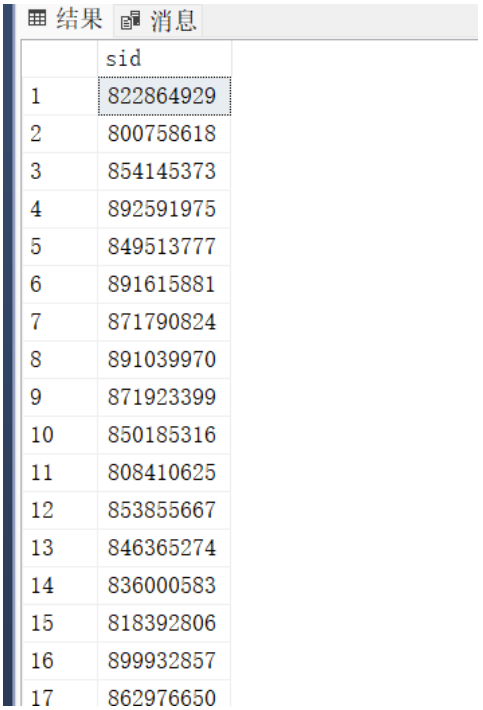
- 注意：查询最大值/最小值，要注意比较两端不要有`null`，因为`null`和任何值比较都是`unknown`，会被where子句过滤掉（判断语句为`IS NULL`和`IS NOT NULL`）
- 结果：

	tid	cid
1	214445507	10004
2	204711560	10003
3	277877392	10047
4	204711560	10029
5	204711560	10019
6	204711560	10024
7	287866460	10003
8	214445507	10012
9	277877392	10025
10	204711560	10016
11	277877392	10027
12	277877392	10021
13	277877392	10041
14	204711560	10001
15	204711560	10026
16	287866460	10005
17	277877392	10047
18	214445507	10022
19	214445507	10050

6. 找出选修课程ERP成绩最高的学生编号;



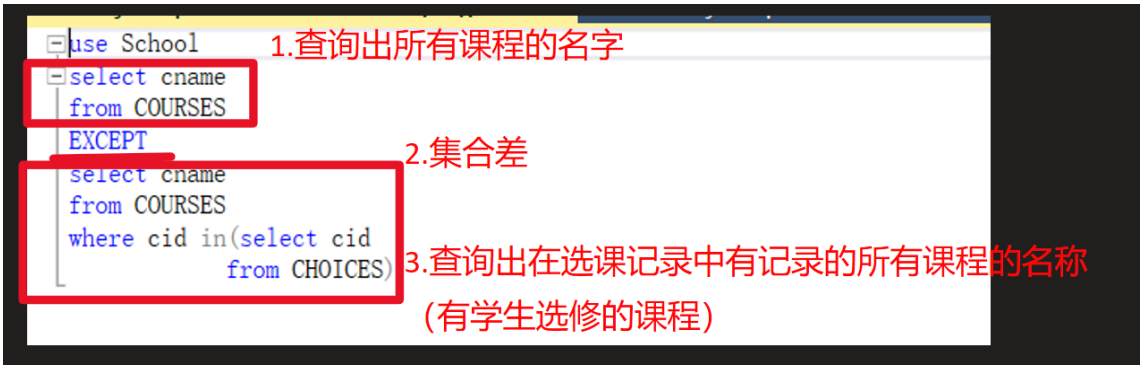
- 注意：比较符两端的`score`不为`null`
- 结果：共95人



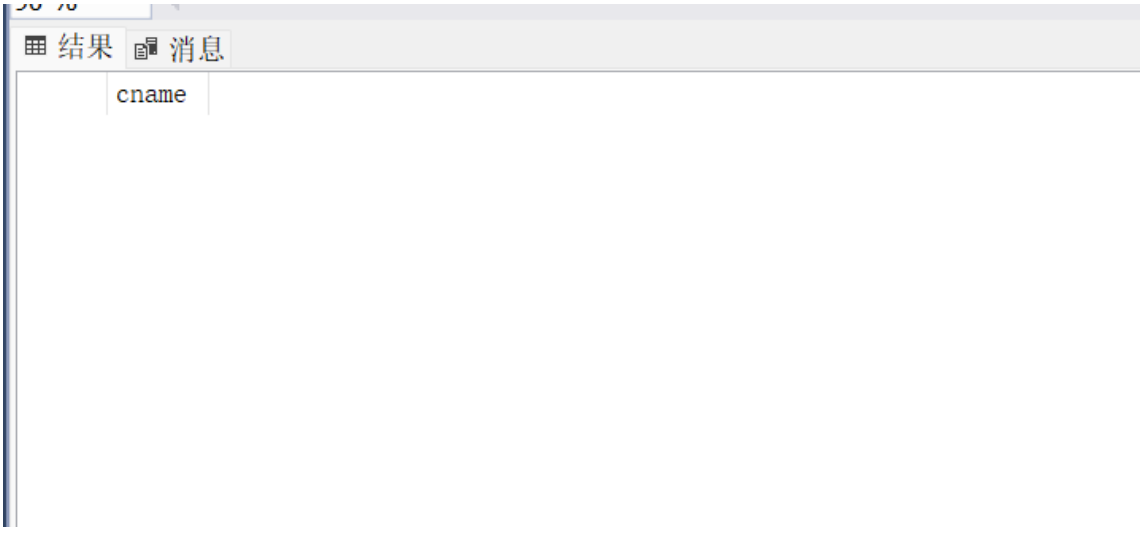
结果 消息

	sid
1	822864929
2	800758618
3	854145373
4	892591975
5	849513777
6	891615881
7	871790824
8	891039970
9	871923399
10	850185316
11	808410625
12	853855667
13	846365274
14	836000583
15	818392806
16	899932857
17	862976650

7. 查询没有学生选修的课程名称



结果：没有没有学生选修过的课程



结果 消息

cname

8. 查询讲授课程UML的教师所讲授的所有课程名称;

```
use School
SELECT distinct cname
from COURSES,CHOICES
where CHOICES.cid=COURSES.cid and CHOICES.tid in (select tid
                                                  from COURSES ,CHOICES
                                                  where COURSES.cid=CHOICES.cid and COURSES.cname='uml')
```

3.可能有多个老师教过同一门课程，需要去重

1.将选课记录和对应的课程具体信息拼接

2. In 查询授课老师的tid取值在子查询的结果关系中的元组

子查询：查询教授过uml课程的老师的tid

总：父查询需要查询教过uml课程的老师的授课记录元组

结果：教过uml的老师教过全部50门课程

90 %

结果 消息

	cname
1	algorithm
2	architectonics
3	artifical intelligence
4	asp
5	basic
6	c
7	c#
8	c++
9	c++/stl programming
10	cobol
11	compiling principle
12	computer graphics
13	computer interface
14	computer network
15	computer storage
16	computer virus
17	corba
18	cpu
19	cryptology

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;

```
use School
SELECT sid
from CHOICES, COURSES
where CHOICES.cid=COURSES.cid and COURSES.cname='database'
INTERSECT
SELECT sid
from CHOICES, COURSES
where CHOICES.cid=COURSES.cid and COURSES.cname='uml'
```

1.查询选修过database的学生的sid

2.集合交

3.查询选修过uml的学生的sid

结果：共303个sid

	sid
285	843875680
286	846776774
287	847324851
288	848578607
289	851346447
290	860124803
291	862053151
292	862244714
293	862515780
294	864990663
295	867225363
296	868291711
297	872278396
298	872468403
299	891620496
300	893390935
301	894506378
302	895662658
303	897958707

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;

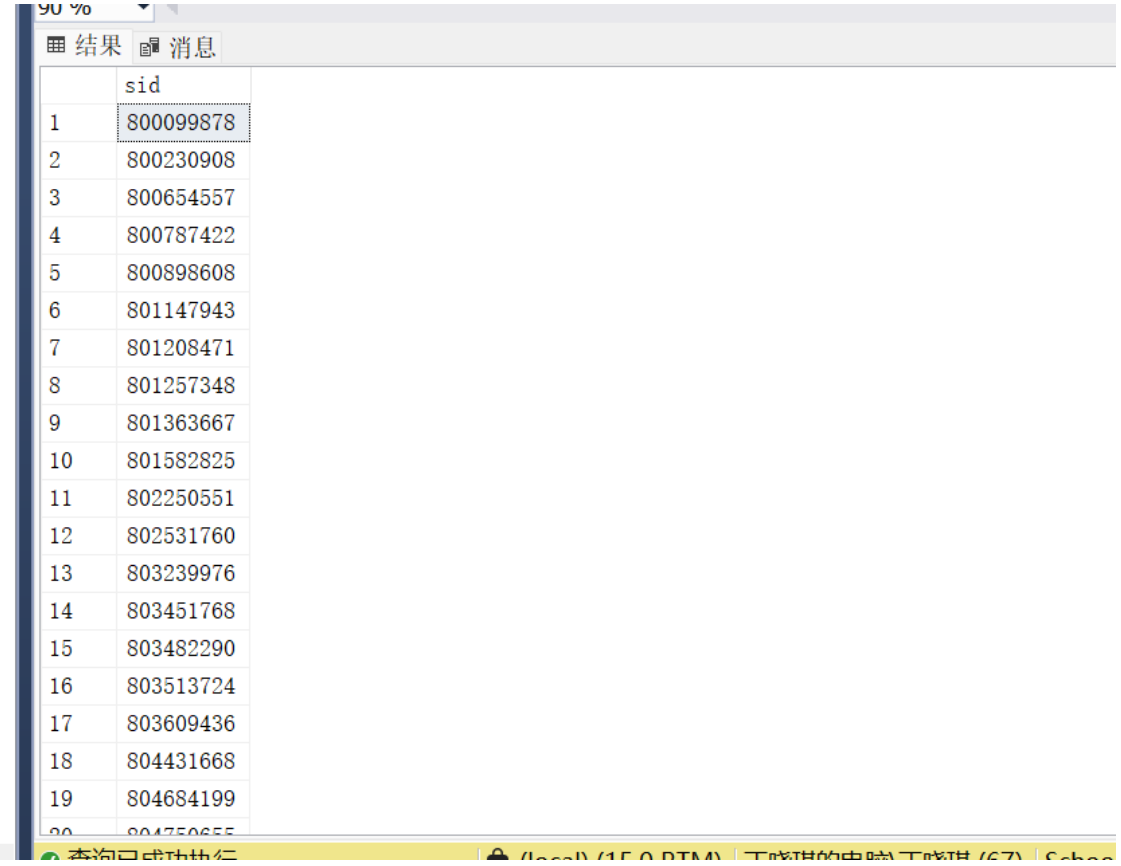
```
use School
SELECT sid
from CHOICES, COURSES
where CHOICES.cid=COURSES.cid and COURSES.cname='database'
EXCEPT
SELECT sid
from CHOICES, COURSES
where CHOICES.cid=COURSES.cid and COURSES.cname='uml'
```

1.查询选修过database的学生的sid

2.集合差

3.查询选修过uml的学生的sid

结果：共5454个元组



	sid
1	800099878
2	800230908
3	800654557
4	800787422
5	800898608
6	801147943
7	801208471
8	801257348
9	801363667
10	801582825
11	802250551
12	802531760
13	803239976
14	803451768
15	803482290
16	803513724
17	803609436
18	804431668
19	804684199

三：实验总结

- 多关系时，要记得拼接，不然就是从多关系的完全笛卡尔积里面查询
- 嵌套查询的子查询中不能使用`order by`子句，外层的查询的属性可以用在子查询中
- 关键字`in`：匹配某个列中的值是否存在于一个给定的集合中
  - 和`= some`和`=ANY`作用相同
  - `not in`作用相反
  - 用法：+子查询，+集合（但是不能直接+关系）
- 关键字`EXISTS`：+子查询，查找是否存在符合条件的元组，存在返回`true`，不然返回`fasle`
- 查询最大/最小：注意比较符两端都不要有`null`

```
>=all+子查询（查找最大）
<=all+子查询（查找最小）
```

- 集合运算：`EXCEPT`集合差，`INTERSECT`集合交，`UNION`集合并
- 带有比较运算符的子查询，该子查询必须返回单值，否则引起编译错误。