

实验二

实验二

数据库描述

实验2.1 数据查询

实验2.2 空值和空集的处理

邓语苏 计科21级本科生

2024/4/3

数据库描述

以School数据库为例，在该数据库中存在四张表格，分别为：

表STUDENT(sid, sname, email, grade);

表TEACHERS(tid, tname, email, salary);

表COURSES(cid, cname, hour);

表CHOICES(no, sid, tid, cid, score)

在数据库中，存在这样的关系：学生可以选择课程，一个课程对应一个教师。在表CHOICES中保存学生的选课记录。

实验2.1 数据查询

按以下要求对数据库进行查询操作：

```
1 | USE School /*切换至School数据库*/
```

(1) 查询年级为2001的所有学生的名称并按编号升序排列。

```
1 | SELECT sname,  
2 |      sid  
3 | FROM STUDENTS  
4 | WHERE grade = 2001  
5 | ORDER BY sid ASC;
```

	score	sid
1	76	823069829
2	87	829348273
3	92	847061074
4	82	860635914
5	77	829785562
6	67	822137137
7	90	826310502
8	60	817636568

(2) 查询学生的选课成绩合格的课程成绩，并把成绩换算为积点（60分对应积点为1，每增加1分，积点增加0.1）。

1. 将SCORE改为浮点型

```
1 ALTER TABLE CHOICES
2 ALTER COLUMN SCORE FLOAT;
```

2. 新建一个名为GPA，类型为浮点型的属性

```
1 ALTER TABLE CHOICES
2 ADD GPA FLOAT;
```

3. 将成绩换算为绩点

```
1 UPDATE CHOICES
2 SET GPA = CASE
3         WHEN score >= 60 THEN ( score - 50 ) / 10
4         ELSE 0
5     END
```

4. 输出结果

```
1 SELECT sname,
2        CHOICES.sid,
3        cid,
4        score,
5        GPA
6 FROM CHOICES,
7 STUDENTS
8 WHERE score >= 60
9        AND CHOICES.sid = STUDENTS.sid
```

5. 复原表格

```
1 ALTER TABLE CHOICES
2 DROP COLUMN GPA
3
4 ALTER TABLE CHOICES
5 ALTER COLUMN SCORE INT
```

	sname	sid	cid	score	GPA
1	pxfys	823069829	10037	76	2.6
2	rfsldreav	829348273	10010	87	3.7
3	qxkbh	847061074	10025	92	4.2
4	xnhdjo	860635914	10039	82	3.2
5	bemgynei	829785562	10028	77	2.7
6	qaxwe	822137137	10011	67	1.7
7	cqkrjkuf	826310502	10005	90	4
8	wzinemrs	817636568	10047	60	1
9	kvamveu	801967882	10021	70	2
10	rekmgdbo	875434315	10048	82	3.2
11	mbfrhwz	830180555	10016	76	2.6
12	kxqlbun	848035070	10007	88	3.8

(3) 查询课时是48或64的课程名称。

```
1 SELECT cname
2 FROM   COURSES
3 WHERE  hour = 48
4        OR hour = 64;
```

	cname
1	computer graphics
2	java
3	design pattern
4	real-time system
5	c
6	computer interface

(4) 查询所有课程名称中含有data的课程编号。

不可用=替代LIKE

```
1 SELECT cname,
2        cid
3 FROM   COURSES
4 WHERE  cname LIKE '%data%';
```

	cname	cid
1	database	10001
2	data structure	10008

(5) 查询所有选课记录的课程号（不重复显示）。

```
1 SELECT DISTINCT cid
2 FROM   CHOICES
```

	cid
1	10008
2	10019
3	10018
4	10040
5	10011
6	10028
7	10035

(6) 统计所有教师的平均工资。

```
1 SELECT Avg(salary) AS avg_salary
2 FROM TEACHERS
```

	avg_salary
1	3417

(7) 查询所有教师的编号及选修其课程的学生们的平均成绩，按平均成绩降序排列。

```
1 SELECT tid,
2        Avg(score) AS avg_score
3 FROM CHOICES
4 GROUP BY tid
5 ORDER BY avg_score DESC
```

	tid	avg_score
1	259380467	90
2	272507136	89
3	241568555	89
4	265299360	88
5	260955336	88
6	266524091	88
7	294109744	88
8	268322984	87
9	225148667	87

(8) 统计各个课程的选课人数和平均成绩。

sid需要剔除重复选课的人

```
1 SELECT cid,
2        Count(DISTINCT sid) AS count_sid,
3        Avg(score) AS avg_score
4 FROM CHOICES
5 GROUP BY CID
```

	cid	count_sid	avg_score
1	10008	5825	75
2	10019	5889	75
3	10018	5812	76
4	10040	5945	75
5	10011	5937	76
6	10028	5886	75
7	10035	5937	76
8	10021	5756	76

(9) 查询至少选修了三门课程的学生编号。

```

1 SELECT sid,
2       Count(CID) AS count_cid
3 FROM   CHOICES
4 GROUP BY SID
5 HAVING Count(CID) >= 3;

```

	sid	count_cid
1	881360462	4
2	822863200	5
3	878053860	4
4	890054584	3
5	888277410	5
6	891044332	3
7	862458436	5
8	839006320	3
9	868752380	5

(10) 查询编号800009026的学生所选的全部课程的课程名和成绩。

```

1 /*正确的写法*/
2 SELECT DISTINCT cname,
3                sid,
4                score
5 FROM   CHOICES
6        INNER JOIN COURSES
7        ON COURSES.cid = CHOICES.cid
8 WHERE  sid = '800009026'
9        AND score IS NOT NULL;

```

	cname	sid	score
1	c	800009026	83
2	tcp/ip protocol	800009026	94

```

1  /*错误的写法*/
2  SELECT DISTINCT cname,
3             sid,
4             score
5  FROM    COURSES,
6          CHOICES
7  WHERE   sid = '800009026'
8          AND score IS NOT NULL

```

	sid	cname	score
1	800009026	database	94
2	800009026	operating system	94
3	800009026	computer graphics	94
4	800009026	java	94
5	800009026	c++	94
6	800009026	design pattern	94
7	800009026	data structure	94
8	800009026	software engineering	94
9	800009026	computer network	94
10	800009026	tcp/ip protocol	94
11	800009026	algorithm	94

(11) 查询所有选修了database的学生的编号。

```

1  /*正确写法 -5757条*/
2  SELECT DISTINCT sid
3  FROM    CHOICES
4          INNER JOIN COURSES
5          ON COURSES.cid = CHOICES.cid
6  WHERE   cname = 'database'

```

5751	823814894
5752	862068694
5753	893531226
5754	874853045
5755	821946666
5756	884448155
5757	857191698

✓ 查询已成功执行。

```

1  /*错误写法 -9w条*/
2  SELECT DISTINCT sid
3  FROM    COURSES,
4          CHOICES
5  WHERE   cname = 'database'

```

99564	854972168	database
99565	840161353	database
99566	849555599	database
99567	874254389	database
99568	862377069	database
99569	842615876	database

(12) 求出选择了同一个课程的学生数。

```

1 SELECT cid,
2     Count(DISTINCT sid) AS count_sid
3 FROM CHOICES
4 GROUP BY cid

```

	cid	count_sid
1	10008	5985
2	10019	6074
3	10018	5969
4	10040	6102
5	10011	6086
6	10028	6042
7	10035	6104
8	10021	5916

(13) 求出至少被两名学生选修的课程编号。

```

1 SELECT cid
2 FROM CHOICES
3 GROUP BY CID
4 HAVING Count(sid) >= 2;

```

	cid
1	10008
2	10019
3	10018
4	10040
5	10011
6	10028
7	10035

(14) 查询选修了编号80009026的学生所选的某个课程的学生编号。

```

1 SELECT sid
2 FROM CHOICES
3 WHERE cid IN (SELECT cid
4               FROM CHOICES
5               WHERE sid = '80009026');

```

空

(15) 查询学生的基本信息及选修课程编号和成绩。

```

1 SELECT DISTINCT CHOICES.sid,
2                 sname,
3                 email,
4                 grade,
5                 cid,
6                 score
7 FROM   CHOICES
8        INNER JOIN STUDENTS
9        ON CHOICES.sid = STUDENTS.sid

```

	sid	sname	email	grade	cid	score
1	800001216	gfxrgs	hhce4@qhldj.gov	1992	10026	67
2	800001216	gfxrgs	hhce4@qhldj.gov	1992	10034	60
3	800002933	vnbgzsvv	pvhxd4l@zqur.org	2002	10008	79
4	800002933	vnbgzsvv	pvhxd4l@zqur.org	2002	10022	60
5	800002933	vnbgzsvv	pvhxd4l@zqur.org	2002	10042	NULL
6	800002933	vnbgzsvv	pvhxd4l@zqur.org	2002	10046	82
7	800005753	waqcj	hlhq0h8@jdba.gov	1992	10005	66
8	800006682	fiiluommh	ihzd6_k@kzvft.gov	1992	10023	94
9	800006682	fiiluommh	ihzd6_k@kzvft.gov	1992	10035	96
10	800006682	fiiluommh	ihzd6_k@kzvft.gov	1992	10045	77

(16) 查询学号850955252的学生的姓名和选修的课程名及成绩。

查询850955252

```

1 SELECT sname,
2        cname,
3        score
4 FROM   CHOICES,
5        STUDENTS,
6        COURSES
7 WHERE  CHOICES.sid = '850955252'
8        AND STUDENTS.sid = CHOICES.sid
9        AND COURSES.cid = CHOICES.cid;

```

sname	cname	score
-------	-------	-------

找到选修课程最多的人的学号


```

1
2 SELECT TOP (1) CHOICES.sid,
3           Count(*) AS count_sid
4 FROM     CHOICES
5           INNER JOIN STUDENTS
6             ON CHOICES.sid = STUDENTS.sid
7           INNER JOIN COURSES
8             ON CHOICES.cid = COURSES.cid
9 GROUP BY CHOICES.sid
10 ORDER BY count_sid DESC

```

	sid	count_sid
1	887150338	5

查询该学生

```

1 SELECT sname,
2        cname,
3        score
4 FROM   CHOICES,
5        STUDENTS,
6        COURSES
7 WHERE  CHOICES.sid = '887150338'
8        AND STUDENTS.sid = CHOICES.sid
9        AND COURSES.cid = CHOICES.cid;

```

	sname	cname	score
1	gszzsizrr	c	50
2	gszzsizrr	real-time system	78
3	gszzsizrr	software engineering	78
4	gszzsizrr	algorithm	NULL
5	gszzsizrr	design pattern	NULL

(17) 查询与学号850955252的学生同年纪的所有学生资料。

```

1 SELECT *
2 FROM   STUDENTS
3 WHERE  grade = (SELECT grade
4                 FROM   STUDENTS
5                 WHERE  sid = '850955252')

```

	grade
1	2001

	sid	sname	email	grade
1	800028044	ztozk	r369l9m@lmykh.gov	2001
2	800041569	pgmrkdhh	xpqi2wc@hrjtp.edu	2001
3	800070739	nkdnfq	pto7n@sci.com	2001
4	800152632	qtxblqzsv	tdvh@mneu.edu	2001
5	800166448	ctvxn	5b6zz@fpplh.org	2001
6	800169970	airnnfv	ytkyo@xeh.org	2001
7	800177146	vaesalave	tla7d4@ypzxr.gov	2001
8	800202438	xiraegdlg	_fow_i@hmbc.edu	2001
9	800268599	dwjny	9pax@ejjk.com	2001
10	800269975	vqjfxfc	i5w9ba@spu.com	2001

(18) 查询所有有选课的学生的详细信息。

有两种查询方式，效果相同

```

1  SELECT *
2  FROM STUDENTS
3  WHERE sid IN(SELECT DISTINCT sid
4                FROM CHOICES)
5
6  SELECT *
7  FROM STUDENTS
8  WHERE EXISTS(SELECT *
9                FROM CHOICES
10               WHERE CHOICES.sid = STUDENTS.sid)

```

	sid	sname	email	grade
1	800001216	gfxrgs	hhce4@qhldj.gov	1992
2	800002933	vn bqzsvv	pvhxd4l@zqur.org	2002
3	800005753	waqcj	hlhq0h8@jdba.gov	1992
4	800006682	fiiluommh	ihzd6_k@kzvft.gov	1992
5	800006941	ogvmu	62sfbd@lrt.gov	1995
6	800007595	uxqqbkjn	cr8g@zrvgt.edu	1997
7	800008565	ehlycg	nach10@uic.com	1999
8	800009026	rcxaihj	4ul4kqb@hko.edu	2002
9	800009099	zapyv	jmqn8@iwaiu.org	1992
10	800009249	zyuoh	8enjrcu@upfw.org	1991

(19) 查询没有学生选的课程编号。

```

1 SELECT cid
2 FROM COURSES
3 WHERE NOT EXISTS(SELECT *
4                   FROM CHOICES
5                   WHERE COURSES.cid = CHOICES.cid)
6
7 SELECT cid
8 FROM COURSES
9 WHERE cid NOT IN(SELECT DISTINCT cid
10                  FROM CHOICES)

```

	cid
1	10004

(20) 查询选修了课程名为C++的课时一样课程名称。

```

1 SELECT sname
2 FROM STUDENTS
3 WHERE sid IN (SELECT sid
4               FROM CHOICES
5               WHERE CHOICES.sid = STUDENTS.sid /*选修了课程*/
6               AND CHOICES.cid IN(SELECT cid
7                                   FROM COURSES x
8                                   WHERE hour = (SELECT hour
9                                                  FROM COURSES y
10                                                 WHERE x.hour =
11                                                  y.hour /*这个课程的课时与C++一样*/
12                                                  AND y.cname =
13                                                  'C++'))))

```

	sname
1	vnbqzsvv
2	waqcj
3	aoaahudi
4	cxjnk
5	efmgfprz
6	ocofw
7	ptqno
8	goqjy
9	hfpmwvs
10	hfxjzne

(21) 找出选修课程成绩最好的选课记录。

```

1 SELECT TOP(1) *
2 FROM CHOICES
3 ORDER BY score DESC

```


无

(25) 利用集合运算，查询选修课程C++或选修课程Java的学生的编号。

```
1 SELECT sid
2 FROM CHOICES
3 WHERE cid IN(SELECT cid
4               FROM COURSES
5               WHERE cname = 'C++')
6 UNION
7 SELECT sid
8 FROM CHOICES
9 WHERE cid IN(SELECT cid
10              FROM COURSES
11              WHERE cname = 'Java')
```

	sid
1	822863200
2	838235261
3	890054584
4	835009071
5	862430162
6	832313883
7	880956250
8	863857955
9	826772208

(26) 实现集合交运算，查询既选修课程C++又选修课程Java的学生的编号。

```
1 SELECT sid
2 FROM CHOICES
3 WHERE cid IN(SELECT cid
4               FROM COURSES
5               WHERE cname = 'C++')
6 intersect
7 SELECT sid
8 FROM CHOICES
9 WHERE cid IN(SELECT cid
10              FROM COURSES
11              WHERE cname = 'Java')
```

无

(27) 实现集合减运算，查询选修课程C++而没有选修课程Java的学生的编号。

```

1 SELECT sid
2 FROM CHOICES
3 WHERE cid IN(SELECT cid
4               FROM COURSES
5               WHERE cname = 'C++')
6 except
7 SELECT sid
8 FROM CHOICES
9 WHERE cid IN(SELECT cid
10              FROM COURSES
11              WHERE cname = 'Java')

```

	sid
1	822863200
2	805425493
3	838235261
4	853379616
5	896729994
6	890054584
7	886557832
8	881768598
9	815570365

实验2.2 空值和空集的处理

(1) 查询所有选课记录的成绩并将它换算为五分制（满分5分，合格3分），注意SCORE取NULL值的情况。

```

1 SELECT no,
2        sid,
3        tid,
4        cid,
5        score / 20 AS gpa
6 FROM CHOICES
7 WHERE score IS NOT NULL

```

	no	sid	tid	cid	gpa
1	500000058	823069829	249596497	10037	3
2	500000253	829348273	202560416	10010	4
3	500000984	850955252	234145610	10021	2
4	500001270	847061074	292043491	10025	4
5	500002143	860635914	238811498	10039	4
6	500002371	829785562	273189968	10028	3
7	500002715	822137137	218922066	10011	3
8	500003262	826310502	267846042	10005	4
9	500004144	817636568	253205179	10047	3

(2) 通过查询选修编号10028的课程的学生的人数，其中成绩合格的学生人数，不合格的学生人数，讨论NULL值的特殊含义。

```

1  /*正确写法*/
2  SELECT cid,
3         Count(CASE
4             WHEN score >= 60 THEN 1
5             END) AS pass_count,
6         Count(CASE
7             WHEN score < 60 THEN 1
8             END) AS fail_count
9  FROM   CHOICES
10 GROUP BY cid
11 HAVING cid = '10028';

```

	cid	pass_count	fail_count
1	10028	4812	755

```

1  /*错误写法：统计了10028课程中各个分数的人数*/
2  SELECT score,
3         Count(CASE
4             WHEN score >= 60 THEN 1
5             END) AS pass_count,
6         Count(CASE
7             WHEN score < 60 THEN 1
8             END) AS fail_count
9  FROM   CHOICES
10 WHERE  cid = '10028'
11 GROUP BY score

```

	score	pass_count	fail_count
1	69	112	0
2	92	117	0
3	75	115	0
4	89	113	0
5	52	0	141
6	95	106	0
7	72	129	0
8	78	115	0
9	66	122	0
10	86	128	0
11	63	103	0
12	55	0	127

(3) 通过实验检验在使用ORDER BY进行排序时，取NULL的项是否出现在结果中？如果有，在什么位置？

```

1 SELECT *
2 FROM CHOICES
3 WHERE cid = '10028'
4 ORDER BY score DESC
5

```

使用升序排序，NULL出现在最后

5563	506645803	885989183	211195523	10028	50
5564	500996294	877038700	226104500	10028	50
5565	502006523	839210834	229617344	10028	50
5566	501509370	851168783	260020156	10028	50
5567	501585861	886648017	274144689	10028	50
5568	501401635	849794066	207634340	10028	NULL
5569	501134428	863295926	291802193	10028	NULL
5570	501064408	819304790	231826353	10028	NULL
5571	501160627	884919470	239793208	10028	NULL
5572	501923231	829320326	232470572	10028	NULL
5573	501750670	812957078	216117368	10028	NULL

使用降序排序，NULL出现在最前

	no	sid	tid	cid	score
1	500410567	840717950	278203144	10028	NULL
2	500481635	898292696	284478828	10028	NULL
3	500809268	846903515	221225913	10028	NULL
4	501064408	819304790	231826353	10028	NULL
5	501134428	863295926	291802193	10028	NULL
6	501160627	884919470	239793208	10028	NULL
7	501401635	849794066	207634340	10028	NULL
8	501750670	812957078	216117368	10028	NULL
9	501923231	829320326	232470572	10028	NULL
10	502380873	836732681	255421693	10028	NULL
11	502389419	858969942	239558382	10028	NULL
12	502559947	805850376	241814811	10028	NULL
13	502662589	824125444	290265401	10028	NULL
14	503149491	877944697	286547481	10028	NULL
15	503156316	875748176	235659888	10028	NULL

(4) 在上面的查询过程中如果加上保留字DISTINCT会有什么效果？

没有使用distinct会展示所有人的成绩，共6k条

```

1 SELECT score
2 FROM CHOICES
3 WHERE cid = '10028'
4 ORDER BY score DESC

```

使用distinct会展示，选10028课程中成绩的不同值，共47条


```

1 SELECT DISTINCT score
2 FROM CHOICES
3 WHERE cid = '10028'
4 ORDER BY score DESC

```

	score
32	68
33	67
34	66
35	65
36	64
37	63
38	62
39	61
40	60
41	55
42	54
43	53
44	52
45	51
46	50
47	NULL

(5) 通过实验说明使用分组GROUP BY对取值为NULL的项的处理。

会将null作为一个单独的项列出

```

1 SELECT score,
2       Count(CASE
3             WHEN score >= 60 THEN 1
4             END) AS pass_count,
5       Count(CASE
6             WHEN score < 60 THEN 1
7             END) AS fail_count,
8       Count(CASE
9             WHEN score IS NULL THEN 1
10            END) AS null_count
11 FROM CHOICES
12 WHERE cid = '10028'
13 GROUP BY score

```

	score	pass_count	fail_count	null_count
27	73	125	0	0
28	85	109	0	0
29	62	131	0	0
30	99	112	0	0
31	76	131	0	0
32	NULL	0	0	475
33	82	115	0	0
34	88	120	0	0
35	53	0	119	0
36	71	124	0	0

(6) 结合分组，使用集合函数求每个同学的平均分、总的选课记录、最高成绩、最低成绩和总成绩。

```

1  /*不考虑NULL，集合函数不会将其一起运算*/
2  SELECT sid,
3         Avg(score) AS avg_score,
4         Count(*) AS records_count,
5         Max(score) AS max_score,
6         Min(score) AS mini_score
7  FROM   CHOICES
8  GROUP BY sid

```

	sid	avg_score	records_count	max_score	mini_score
1	881360462	78	4	96	50
2	822863200	82	5	92	71
3	826765496	99	1	99	99
4	878053860	64	4	70	54
5	890054584	77	3	99	51
6	888277410	73	5	90	65
7	865375956	73	2	79	67
8	821819409	98	1	98	98
9	835593153	93	2	93	93
10	891044332	93	3	99	85

```

1  /*考虑NULL值，将NULL替换为0*/
2  SELECT sid,
3         Avg(isnull(score,0)) AS avg_score,
4         Count(*) AS records_count,
5         Max(isnull(score,0)) AS max_score,
6         Min(isnull(score,0)) AS mini_score
7  FROM   CHOICES
8  GROUP BY sid

```

	sid	avg_score	records_count	max_score	mini_score
1	881360462	58	4	96	0
2	822863200	82	5	92	71
3	826765496	99	1	99	99
4	878053860	64	4	70	54
5	890054584	77	3	99	51
6	888277410	73	5	90	65
7	865375956	73	2	79	67
8	821819409	98	1	98	98
9	835593153	46	2	93	0
10	891044332	93	3	99	85

(7) 查询成绩小于60的选课记录，统计总数、平均分、最大值和最小值。

```

1 select
2     Count(*) AS fail_records_count,
3     Avg(score) AS avg_score,
4     Max(score) AS max_score,
5     Min(score) AS mini_score
6 from CHOICES
7 where score<60

```

	fail_records_count	avg_score	max_score	mini_score
1	35293	52	55	50

(8) 采用嵌套查询的方式，利用比较运算符和谓词ALL的结合来查询表COURSES中最少的课时。假设数据库中只有一个记录的时候，使用前面的方法会得到什么结果，为什么？

```

1 select distinct hour
2 from COURSES
3 where hour<=all(select hour
4 from COURSES)

```

	hour
1	48

如果数据库中只有一个记录，那么子查询SELECT hour FROM COURSES将返回该记录的课时，并且它是该表课时的最小值。因此，主查询中WHERE子句的条件 hour <= ALL(SELECT hour FROM COURSES)也将成立，因此查询结果将为该记录的课时值。

(9) 创建一个学生表S (NO, SID, SNAME)，教师表T (NO, TID, TNAME) 作为实验用的表。其中NO分别是这两个表的主键，其他键允许为空。

向S插入元组 (1, 0129871001, 王小明)、(2, 0129871002, 李兰)、(3, 0129871005, NULL)、(4, 0129871004, 关红)；

向T插入元组1, 100189, 王小明)、(2, 100180, 李小)、(3, 100121, NULL)、(4, 100128, NULL)。

对这两个表作对姓名的等值连接运算，找出既是老师又是学生的人员的学生编号和老师编号。

```

1 CREATE TABLE S(

```

```

2 NO INT PRIMARY KEY,
3 SID CHAR(10),
4 SNAME CHAR(10)
5 )
6 INSERT INTO S(NO,SID,SNAME)VALUES(1, '0129871001','王小明');
7 INSERT INTO S(NO,SID,SNAME)VALUES(2, '0129871002','李兰');
8 INSERT INTO S(NO,SID,SNAME)VALUES(3, '0129871005',NULL);
9 INSERT INTO S(NO,SID,SNAME)VALUES(4, '0129871004','关红');
10 CREATE TABLE T(
11 NO INT PRIMARY KEY,
12 TID CHAR(10),
13 TNAME CHAR(10)
14 )
15 INSERT INTO T(NO,TID,TNAME)VALUES(1, '100189','王小明');
16 INSERT INTO T(NO,TID,TNAME)VALUES(2, '100180','李小');
17 INSERT INTO T(NO,TID,TNAME)VALUES(3, '100121',NULL);
18 INSERT INTO T(NO,TID,TNAME)VALUES(4, '100128',NULL);
19 select *
20 from S;
21 select *
22 from T;

```

	NO	SID	SNAME
1	1	0129871001	王小明
2	2	0129871002	李兰
3	3	0129871005	NULL
4	4	0129871004	关红

	NO	TID	TNAME
1	1	100189	王小明
2	2	100180	李小
3	3	100121	NULL
4	4	100128	NULL

```

1
2
3 select SID,TID,SNAME
4 from S
5 join T on SNAME=TNAME
6 where SNAME is not null and TNAME is not null;

```

	SID	TID	SNAME
1	0129871001	100189	王小明