

mysql查询练习

新建一个查询用的数据库:selectTest

```
CREATE DATABASE selectTest;
```

选择该数据库:

```
USE selectTest;
```

学生表:

student

学号

姓名

性别

出生日期

所在班级

```
CREATE TABLE student(  
    s_no VARCHAR(20) PRIMARY KEY COMMENT'学生学号',  
    s_name VARCHAR(20) NOT NULL COMMENT'学生姓名 不能为空',  
    s_sex VARCHAR(10) NOT NULL COMMENT'学生性别',  
    s_birthday DATETIME COMMENT'学生生日',  
    s_class VARCHAR(20) COMMENT'学生所在的班级'  
);
```

课程表:

course

课程号

课程课程名称

教师编号

```
CREATE TABLE course(  
    c_no VARCHAR(20) PRIMARY KEY COMMENT'课程号',  
    c_name VARCHAR(20) NOT NULL COMMENT'课程名称',  
    t_no VARCHAR(20) NOT NULL COMMENT'教师编号 外键关联teacher表',  
    FOREIGN KEY(t_no) references teacher(t_no)  
);
```

报错:Failed to open the referenced table 'teacher'
我们先创建teacher表

教师表

teacher

教师编号

教师名字

教师性别

出生日期

职称

所在部门

```
CREATE TABLE teacher(  
    t_no VARCHAR(20) PRIMARY KEY COMMENT'教师编号',  
    t_name VARCHAR(20) NOT NULL COMMENT'教师姓名',  
    t_sex VARCHAR(20) NOT NULL COMMENT'教师性别',  
    t_birthday DATETIME COMMENT'教师生日',  
    t_rof VARCHAR(20) NOT NULL COMMENT'教师职称',  
    t_depart VARCHAR(20) NOT NULL COMMENT'教师所在的部门'  
);
```

成绩表

score

学号

课程号
成绩

注意:视频中原先只有一个主键s_no ,后来修改了

```
CREATE TABLE score (  
    s_no VARCHAR(20) NOT NULL COMMENT '成绩表的编号 依赖学生学号',  
    c_no VARCHAR(20) NOT NULL COMMENT '课程号 依赖于课程表中的c_id',  
    sc_degree decimal,  
    foreign key(s_no) references student(s_no),  
    foreign key(c_no) references course(c_no),  
    PRIMARY KEY(s_no,c_no)  
);
```

查看创建的表以及架构
SHOW TABLES;

Tables_in_selecttest
course
score
student
teacher

查看student表结构 DESCRIBE student;

Field	Type	Null	Key	Default	Extra
s_no	varchar(20)	NO	PRI	NULL	
s_name	varchar(20)	NO		NULL	
s_sex	varchar(10)	NO		NULL	
s_birthday	datetime	YES		NULL	
s_class	varchar(20)	YES		NULL	

查看teacher表结构 DESCRIBE teacher;

Field	Type	Null	Key	Default	Extra
t_no	varchar(20)	NO	PRI	NULL	
t_name	varchar(20)	NO		NULL	
t_sex	varchar(20)	NO		NULL	
t_birthday	datetime	YES		NULL	
t_rof	varchar(20)	NO		NULL	
t_depart	varchar(20)	NO		NULL	

查看course表结构 DESCRIBE course;

Field	Type	Null	Key	Default	Extra
c_no	varchar(20)	NO	PRI	NULL	
c_name	varchar(20)	NO		NULL	
t_no	varchar(20)	NO	MUL	NULL	

查看score表结构 DESCRIBE score;

Field	Type	Null	Key	Default	Extra
s_no	varchar(20)	NO	PRI	NULL	
c_no	varchar(20)	NO	MUL	NULL	
sc_degree	decimal(10,0)	YES		NULL	

向表中添加数据

```
-- 学生表数据
```

```
INSERT INTO student VALUES('101','曾华','男','1977-09-01','95033');
INSERT INTO student VALUES('102','匡明','男','1975-10-02','95031');
INSERT INTO student VALUES('103','王丽','女','1976-01-23','95033');
INSERT INTO student VALUES('104','李军','男','1976-02-20','95033');
INSERT INTO student VALUES('105','王芳','女','1975-02-10','95031');
INSERT INTO student VALUES('106','陆军','男','1974-06-03','95031');
INSERT INTO student VALUES('107','王尼玛','男','1976-02-20','95033');
INSERT INTO student VALUES('108','张全蛋','男','1975-02-10','95031');
INSERT INTO student VALUES('109','赵铁柱','男','1974-06-03','95031');

-- 教师表数据
INSERT INTO teacher VALUES('804','李诚','男','1958-12-02','副教授','计算机系');
INSERT INTO teacher VALUES('856','张旭','男','1969-03-12','讲师','电子工程系');
INSERT INTO teacher VALUES('825','王萍','女','1972-05-05','助教','计算机系');
INSERT INTO teacher VALUES('831','刘冰','女','1977-08-14','助教','电子工程系');

-- 添加课程表
INSERT INTO course VALUES('3-105','计算机导论','825');
INSERT INTO course VALUES('3-245','操作系统','804');
INSERT INTO course VALUES('6-166','数字电路','856');
INSERT INTO course VALUES('9-888','高等数学','831');

-- 添加成绩表
INSERT INTO score VALUES('103','3-245','86');
INSERT INTO score VALUES('105','3-245','75');
INSERT INTO score VALUES('109','3-245','68');
INSERT INTO score VALUES('103','3-105','92');

INSERT INTO score VALUES('105','3-105','88');
INSERT INTO score VALUES('109','3-105','76');
INSERT INTO score VALUES('103','6-166','85');

INSERT INTO score VALUES('105','6-166','79');
INSERT INTO score VALUES('109','6-166','81');
```

几张表的数据展现

student SELECT * FROM student;

s_no	s_name	s_sex	s_birthday	s_class
101	曾华	男	1977-09-01 00:00:00	95033
102	匡明	男	1975-10-02 00:00:00	95031
103	王丽	女	1976-01-23 00:00:00	95033
104	李军	男	1976-02-20 00:00:00	95033
105	王芳	女	1975-02-10 00:00:00	95031
106	陆军	男	1974-06-03 00:00:00	95031
107	王尼玛	男	1976-02-20 00:00:00	95033
108	张全蛋	男	1975-02-10 00:00:00	95031
109	赵铁柱	男	1974-06-03 00:00:00	95031

teacher SELECT * FROM teacher;

t_no	t_name	t_sex	t_birthday	t_rof	t_depart
804	李诚	男	1958-12-02 00:00:00	副教授	计算机系
825	王萍	女	1972-05-05 00:00:00	助教	计算机系
831	刘冰	女	1977-08-14 00:00:00	助教	电子工程系
856	张旭	男	1969-03-12 00:00:00	讲师	电子工程系

score SELECT * FROM score;

s_no	c_no	sc_degree
103	3-105	92
103	3-245	86
103	6-166	85
105	3-105	88

105	3-245	75
105	6-166	79
109	3-105	76
109	3-245	68
109	6-166	81

course SELECT * FROM course;

c_no	c_name	t_no
3-105	计算机导论	825
3-245	操作系统	804
6-166	数字电路	856
9-888	高等数学	831

查询联系

-- 1.查询student表中所有的记录

SELECT * FROM student;

s_no	s_name	s_sex	s_birthday	s_class
1	admin	男	1999-09-09 00:00:00	95033
101	曾华	男	1977-09-01 00:00:00	95033
102	匡明	男	1975-10-02 00:00:00	95031
103	王丽	女	1976-01-23 00:00:00	95033
104	李军	男	1976-02-20 00:00:00	95033
105	王芳	女	1975-02-10 00:00:00	95031
106	陆军	男	1974-06-03 00:00:00	95031
107	王尼玛	男	1976-02-20 00:00:00	95033
108	张全蛋	男	1975-02-10 00:00:00	95031
109	赵铁柱	男	1974-06-03 00:00:00	95031

-- 2.查询student表中所有记录的s_name,s_sex和s_class列

SELECT s_no,s_name,s_class FROM student;

s_no	s_name	s_class
1	admin	95033
101	曾华	95033
102	匡明	95031
103	王丽	95033
104	李军	95033
105	王芳	95031
106	陆军	95031
107	王尼玛	95033
108	张全蛋	95031
109	赵铁柱	95031

-- 3.查询教师所有的单位但是不重复的t_depart列

SELECT distinct (t_depart) FROM teacher;

t_depart
计算机系
计算机系

| 电子工程系 |
+-----+

-- 4.查询score表中成绩在60-80之间所有的记录(sc_degree)

注意:BETWEEN... AND... 是包含边界的

```
SELECT * FROM score WHERE sc_degree BETWEEN 61 AND 79;
SELECT * FROM score WHERE sc_degree < 80 AND sc_degree > 60 ;
```

s_no	c_no	sc_degree
105	3-245	75
105	6-166	79
109	3-105	76
109	3-245	68

-- 5.查询score表中成绩为85, 86, 或者88的记录(sc_degree)

```
SELECT * FROM score WHERE sc_degree IN(85, 86, 88);
```

s_no	c_no	sc_degree
103	3-245	86
103	6-166	85
105	3-105	88

-- 6.查询student表中'95031'班或者性别为'女'的同学记录

```
SELECT * FROM student WHERE s_class = '95031' OR s_sex = '女';
```

s_no	s_name	s_sex	s_birthday	s_class
102	匡明	男	1975-10-02 00:00:00	95031
103	王丽	女	1976-01-23 00:00:00	95033
105	王芳	女	1975-02-10 00:00:00	95031
106	陆军	男	1974-06-03 00:00:00	95031
108	张全蛋	男	1975-02-10 00:00:00	95031
109	赵铁柱	男	1974-06-03 00:00:00	95031

-- 7.以class降序查询student表中所有的记录

```
SELECT * FROM student ORDER BY s_class desc;
```

s_no	s_name	s_sex	s_birthday	s_class
1	admin	男	1999-09-09 00:00:00	95033
101	曾华	男	1977-09-01 00:00:00	95033
103	王丽	女	1976-01-23 00:00:00	95033
104	李军	男	1976-02-20 00:00:00	95033
107	王尼玛	男	1976-02-20 00:00:00	95033
102	匡明	男	1975-10-02 00:00:00	95031
105	王芳	女	1975-02-10 00:00:00	95031
106	陆军	男	1974-06-03 00:00:00	95031
108	张全蛋	男	1975-02-10 00:00:00	95031
109	赵铁柱	男	1974-06-03 00:00:00	95031

-- 8.以c_no升序,sc_degree降序插叙score表中所有的数据
先以c_no进行升序,若c_no相同,则以sc_degree降序

```
SELECT * FROM score ORDER BY c_no ASC,sc_degree DESC;
```

s_no	c_no	sc_degree
103	3-105	92
105	3-105	88
109	3-105	76
103	3-245	86
105	3-245	75
109	3-245	68
103	6-166	85
109	6-166	81
105	6-166	79

-- 9.查询'95031'班的学生人数

```
SELECT COUNT(s_no) FROM student WHERE s_class = '95031';
```

COUNT(s_no)
5

-- 10.查询score表中的最高分数的学生号和课程号.(子查询或者排序)

--查询步骤

1.先找到最高分

最高分: SELECT MAX(sc_degree) FROM score;

2.通过我们找到的最高分的分数来从score中找到我们需要的学生号和课程号

SELECT c_no, s_no FROM score WHERE sc_degree = (最高分)

```
SELECT c_no, s_no FROM score WHERE sc_degree = (SELECT MAX(sc_degree) FROM score);
```

c_no	s_no
3-105	103

排序 最高分有多个的情况下可能有数据问题

limit x,y (x:表示从X条数据开始 y:需要查出多少条)

SELECT c_no, s_no FROM score ORDER BY sc_degree DESC LIMIT 0,1; 这个OK

但是我们在插入一条数据:

```
INSERT INTO score VALUES('101','9-888','92');
```

再用排序法去查得到:

c_no	s_no
9-888	101

有两条数据但是只显示一条,有问题

若我们用子查询的方法来查询的话会得到:

c_no	s_no
9-888	101
3-105	103

完全没有问题.

最后为了和视频数据一致删除刚刚插入的数据:

```
DELETE FROM score WHERE c_no = '9-888' AND s_no = '101';
```

-- 11.查询每门课的平均成绩

```
SELECT c_no,AVG(sc_degree) FROM SCORE GROUP BY c_no;
```

c_no	AVG(sc_degree)
3-105	87.6667
3-245	76.3333
6-166	81.6667

-- 12.查询score表中至少有2名学生选修的,并且以3开头的课程的平均分

```
SELECT AVG(sc_degree),c_no from score GROUP BY c_no HAVING COUNT(c_no) >= 2 AND c_no like '3%' ;
```

AVG(sc_degree)	c_no
85.3333	3-105
76.3333	3-245

-- 13.查询分数大于70但是小于90的s_no列:

```
SELECT s_no,sc_degree FROM score WHERE sc_degree BETWEEN 71 AND 89;
```

s_no	sc_degree
103	86
103	85
105	88
105	75
105	79
109	76
109	81

进阶:显示s_name,c_name

```
select s_name,sc_degree,c_name FROM score,student,course WHERE score.s_no = student.s_no AND score.c_no = course.c_no AND sc_degree BETWEEN 71 AND 89;
```

s_name	sc_degree	c_name
王丽	86	操作系统
王丽	85	数字电路
王芳	88	计算机导论
王芳	75	操作系统
王芳	79	数字电路
赵铁柱	76	计算机导论
赵铁柱	81	数字电路

-- 14.查询所有的学生 s_name , c_no, sc_degree列

```
SELECT s_name, c_no, sc_degree FROM student,score WHERE student.s_no = score.s_no;
```

s_name	c_no	sc_degree
王丽	3-105	92
王丽	3-245	86
王丽	6-166	85
王芳	3-105	88
王芳	3-245	75
王芳	6-166	79
赵铁柱	3-105	76
赵铁柱	3-245	68
赵铁柱	6-166	81

-- 15.查询所有学生的s_no, c_name, sc_degree列

```
SELECT student.s_no, c_name, sc_degree FROM student, course, score WHERE student.s_no = score.s_no
AND score.c_no = course.c_no ;
```

s_no	c_name	sc_degree
103	计算机导论	92
103	操作系统	86
103	数字电路	85
105	计算机导论	88
105	操作系统	75
105	数字电路	79
109	计算机导论	76
109	操作系统	68
109	数字电路	81

-- 16.查询所有的学生 s_name , c_name, sc_degree列

将上面的c_no 的值换位c_name

```
SELECT s_name, c_name, sc_degree FROM student, course, score WHERE student.s_no = score.s_no AND
score.c_no = course.c_no;
```

s_name	c_name	sc_degree
王丽	计算机导论	92
王丽	操作系统	86
王丽	数字电路	85
王芳	计算机导论	88
王芳	操作系统	75
王芳	数字电路	79
赵铁柱	计算机导论	76
赵铁柱	操作系统	68
赵铁柱	数字电路	81

-- 17.查询班级是'95031'班学生每门课的平均分

```
select c_no,AVG(sc_degree) from score WHERE s_no IN (select s_no from student where s_class = '95031')
GROUP BY c_no;
```

```
select c_no,AVG(sc_degree) from score AS sc LEFT JOIN student AS s ON sc.s_no = s.s_no WHERE s.s_class
= '95031' GROUP BY sc.c_no;
```

```
SELECT sc.c_no,AVG(sc.sc_degree) FROM student AS s, score AS SC WHERE s.s_class = '95031' AND s.s_no =
sc.s_no GROUP BY sc.c_no ;
```

c_no	AVG(sc.sc_degree)
3-105	82.0000
3-245	71.5000
6-166	80.0000

进阶,加入课程名称:

```
SELECT sc.c_no, c.c_name, AVG(sc.sc_degree) FROM student AS s, score AS SC, course AS c WHERE
s.s_class = '95031' AND s.s_no = sc.s_no AND c.c_no = sc.c_no GROUP BY sc.c_no ;
```

c_no	c_name	AVG(sc.sc_degree)
3-105	计算机导论	82.0000
3-245	操作系统	71.5000
6-166	数字电路	80.0000

-- 18.查询选修"3-105"课程的成绩高于'109'号同学'3-105'成绩 的所有同学的记录

(在大家都在选修3-105的背景下 查询 所有 分数 比 学号为"109"还要高的学生信息)

```
SELECT * FROM student AS s, score AS sc WHERE sc.c_no = '3-105' AND sc.sc_degree > (SELECT sc_degree
FROM score WHERE s_no = '109' AND c_no = '3-105') AND s.s_no = sc.s_no;
```

s_no	s_name	s_sex	s_birthday	s_class	s_no	c_no	sc_degree
103	王丽	女	1976-01-23 00:00:00	95033	103	3-105	92
105	王芳	女	1975-02-10 00:00:00	95031	105	3-105	88

-- 19.查询成绩高于学号为'109',课程号为'3-105'的成绩的所有记录

```
SELECT * FROM score WHERE sc_degree > (SELECT sc_degree FROM score WHERE s_no = '109' AND c_no = '3-105');
```

s_no	c_no	sc_degree
103	3-105	92
103	3-245	86
103	6-166	85
105	3-105	88
105	6-166	79
109	6-166	81

不过视频中仅仅查出来了score记录,但是并没有学生的信息,按照上面的来修改:

```
SELECT * FROM student AS s, score AS sc WHERE sc.sc_degree > (SELECT sc_degree FROM score WHERE s_no =
'109' AND c_no = '3-105') AND s.s_no = sc.s_no;
```

还可以再进一步:

```
SELECT s.s_no AS'学生学号', s.s_name AS'学生姓名', s.sex AS'性别', s.class AS'班级', c.c_no AS'课程编号',
c.c_name AS'课程名称', sc.sc_degree AS'分数' FROM student AS s, score AS sc ,course AS c WHERE
sc.sc_degree > (SELECT sc_degree FROM score WHERE s_no = '109' AND c_no = '3-105') AND s.s_no = sc.s_no
AND sc.c_no = c.c_no;
```

学生学号	学生姓名	性别	班级	课程编号	课程名称	分数
103	王丽	女	95033	3-105	计算机导论	92
103	王丽	女	95033	3-245	操作系统	86
103	王丽	女	95033	6-166	数字电路	85
105	王芳	女	95031	3-105	计算机导论	88
105	王芳	女	95031	6-166	数字电路	79
109	赵铁柱	男	95031	6-166	数字电路	81

-- 20.查询所有学号为108.101的同学同年出生的所有学生的s_no,s_name和s_birthday

```
SELECT * FROM student WHERE YEAR(s_birthday) IN (SELECT YEAR(s_birthday) FROM student WHERE s_no
IN('108','101'));
```

s_no	s_name	s_sex	s_birthday	s_class
101	曾华	男	1977-09-01 00:00:00	95033
102	匡明	男	1975-10-02 00:00:00	95031
105	王芳	女	1975-02-10 00:00:00	95031
108	张全蛋	男	1975-02-10 00:00:00	95031

-- 21.查询 张旭 教师任课的学生的成绩

```
select * from student where s_no IN (SELECT s_no FROM score WHERE c_no = (SELECT c_no FROM course WHERE t_no = (SELECT t_no FROM teacher WHERE t_name='张旭')));
```

s_no	s_name	s_sex	s_birthday	s_class
103	王丽	女	1976-01-23 00:00:00	95033
105	王芳	女	1975-02-10 00:00:00	95031
109	赵铁柱	男	1974-06-03 00:00:00	95031

-- 22.查询选修课程的同学人数多余 5 人的教师姓名

```
SELECT * FROM teacher WHERE t_no IN(SELECT t_no FROM course WHERE c_no IN (SELECT c_no FROM score GROUP BY c_no HAVING COUNT(s_no) > 5));
```

Empty set (0.00 sec) 没有符合条件的

为了效果,添加数据:

```
INSERT INTO score VALUES('101','3-105','90');
INSERT INTO score VALUES('102','3-105','91');
INSERT INTO score VALUES('104','3-105','89');
```

再次查询可得:

t_no	t_name	t_sex	t_birthday	t_rof	t_depart
825	王萍	女	1972-05-05 00:00:00	助教	计算机系

注意:视频中用 '=' 是不严谨的,实际中你根本不知道有多少条件是符合的,要用IN

-- 23.查询95033班和95031班全体学生的记录

```
SELECT * FROM student WHERE s_class IN('95031','95033') ORDER BY s_class ;
```

s_no	s_name	s_sex	s_birthday	s_class
102	匡明	男	1975-10-02 00:00:00	95031
105	王芳	女	1975-02-10 00:00:00	95031
106	陆军	男	1974-06-03 00:00:00	95031
108	张全蛋	男	1975-02-10 00:00:00	95031
109	赵铁柱	男	1974-06-03 00:00:00	95031
101	曾华	男	1977-09-01 00:00:00	95033
103	王丽	女	1976-01-23 00:00:00	95033
104	李军	男	1976-02-20 00:00:00	95033
107	王尼玛	男	1976-02-20 00:00:00	95033

由于视频中就只有这两个班,所以要插入数据:

```
INSERT INTO student VALUES('110','张飞','男','1974-06-03','95038');
```

再次查询,还是上面那些

我在写的时候,写成了:查询95033班和95031班全体学生每门课的成绩以及负责该课程的老师,最后以class来排序

写都写了,那就放出来:

```
SELECT s.s_no, s.s_name, s.s_birthday, s.s_class, c.c_no, c.c_name, sc.sc_degree, t.t_name FROM student AS s, course AS c, score AS sc, teacher AS t WHERE s.s_class IN('95031','95033') AND s.s_no = sc.s_no AND sc.c_no = c.c_no AND c.t_no = t.t_no;
```

s_no	s_name	s_birthday	s_class	c_no	c_name	sc_degree	t_name
102	匡明	1975-10-02 00:00:00	95031	3-105	计算机导论	91	王萍
105	王芳	1975-02-10 00:00:00	95031	3-105	计算机导论	88	王萍
105	王芳	1975-02-10 00:00:00	95031	3-245	操作系统	75	李诚
105	王芳	1975-02-10 00:00:00	95031	6-166	数字电路	79	张旭
109	赵铁柱	1974-06-03 00:00:00	95031	3-105	计算机导论	76	王萍
109	赵铁柱	1974-06-03 00:00:00	95031	3-245	操作系统	68	李诚
109	赵铁柱	1974-06-03 00:00:00	95031	6-166	数字电路	81	张旭
101	曾华	1977-09-01 00:00:00	95033	3-105	计算机导论	90	王萍

103	王丽	1976-01-23 00:00:00	95033	3-105	计算机导论	92	王萍
103	王丽	1976-01-23 00:00:00	95033	3-245	操作系统	86	李诚
103	王丽	1976-01-23 00:00:00	95033	6-166	数字电路	85	张旭
104	李军	1976-02-20 00:00:00	95033	3-105	计算机导论	89	王萍

-- 24.查询存在85分以上成绩的课程c_no

```
SELECT * FROM score where sc_degree > 85;
```

s_no	c_no	sc_degree
101	3-105	90
102	3-105	91
103	3-105	92
103	3-245	86
104	3-105	89
105	3-105	88

升级版:知道c_no还不行,我们还需要看到c_name 以及对应的老师

```
SELECT sc.c_no,c.c_name, t.t_name FROM score AS sc, course AS c, teacher AS t WHERE sc.c_no IN(SELECT c_no FROM score where sc_degree > 85) AND sc.c_no = c.c_no AND c.t_no = t.t_no GROUP BY c.c_name;
```

c_no	c_name	t_name
3-105	计算机导论	王萍
3-245	操作系统	李诚

这两位老师教出来的学生都有85分以上的

-- 25.查出所有'计算机系' 教师所教课程的成绩表

```
SELECT * FROM score WHERE c_no IN (SELECT c_no FROM course WHERE t_no IN (SELECT t_no FROM teacher WHERE t_depart = '计算机系'));
```

s_no	c_no	sc_degree
103	3-245	86
105	3-245	75
109	3-245	68
101	3-105	90
102	3-105	91
103	3-105	92
104	3-105	89
105	3-105	88
109	3-105	76

进阶:

```
SELECT t.t_name,t.t_depart,c.c_name,s.s_name,s_class,sc.sc_degree FROM course AS c, score AS sc, teacher AS t,student AS s WHERE c.t_no IN (select t_no FROM teacher WHERE t_depart = '计算机系') AND c.t_no = t.t_no AND c.c_no = sc.c_no AND sc.s_no = s.s_no ;
```

教师名称	部门	课程名称	学生名称	班级	分数
t_name	t_depart	c_name	s_name	s_class	sc_degree
李诚	计算机系	操作系统	王丽	95033	86
李诚	计算机系	操作系统	王芳	95031	75
李诚	计算机系	操作系统	赵铁柱	95031	68
王萍	计算机系	计算机导论	曾华	95033	90
王萍	计算机系	计算机导论	匡明	95031	91
王萍	计算机系	计算机导论	王丽	95033	92
王萍	计算机系	计算机导论	李军	95033	89
王萍	计算机系	计算机导论	王芳	95031	88
王萍	计算机系	计算机导论	赵铁柱	95031	76

-- 26.查询'计算机系'与'电子工程系'不同职称的教师的name和rof

```
SELECT * FROM teacher WHERE t_depart = '计算机系' AND t_rof NOT IN (SELECT t_rof FROM teacher WHERE t_depart = '电子工程系')
UNION
SELECT * FROM teacher WHERE t_depart = '电子工程系' AND t_rof NOT IN (SELECT t_rof FROM teacher WHERE t_depart = '计算机系');
```

t_no	t_name	t_sex	t_birthday	t_rof	t_depart
804	李诚	男	1958-12-02 00:00:00	副教授	计算机系
856	张旭	男	1969-03-12 00:00:00	讲师	电子工程系

-- 27, 查询选修编号为"3-105"课程且成绩至少高于选修编号为'3-245'同学的c_no,s_no和sc_degree,并且按照sc_degree从高到地次序排序

```
select * from score where c_no = '3-105' AND sc_degree > ANY(SELECT sc_degree FROM score WHERE c_no = '3-245' ) ORDER BY sc_degree desc ;
```

s_no	c_no	sc_degree
103	3-105	92
102	3-105	91
101	3-105	90
104	3-105	89
105	3-105	88
109	3-105	76

-- 28.查询选修编号为"3-105"且成绩高于选修编号为"3-245"课程的同学c_no.s_no和sc_degree

```
SELECT * FROM score WHERE sc_degree > ALL (select sc_degree from score WHERE c_no = '3-245') AND c_no = '3-105';
```

s_no	c_no	sc_degree
101	3-105	90
102	3-105	91
103	3-105	92
104	3-105	89
105	3-105	88

进阶:查出学生的信息,课程名称,分数(s_name c_name,sc_degree)

```
SELECT s.s_name , c.c_name ,sc.sc_degree FROM score AS sc, student AS s,course AS c WHERE sc_degree > ALL (select sc_degree from score WHERE c.c_no = '3-245') AND c.c_no = '3-105' AND sc.s_no = s.s_no AND sc.c_no = c.c_no ;
```

s_name	c_name	sc_degree
曾华	计算机导论	90
匡明	计算机导论	91
王丽	计算机导论	92
李军	计算机导论	89
王芳	计算机导论	88
赵铁柱	计算机导论	76

总结: ANY 和 ALL

ANY:表示任何一个就行了,如:数组A中的值比数组B中任何一个都要大,那么只要A和B中最小的比较就行了.

ALL:表示所有都要比较,如:数组A中的值比数组B中所有的数都要大,那么A要和B中最大的值比较才行.

-- 29. 查询所有教师和同学的名字,sex, birthday

```
SELECT s_name AS name, s_sex AS sex, s_birthday AS birthday FROM student
UNION
SELECT t_name AS name, t_sex AS sex, t_birthday AS birthday FROM teacher;
```

name	sex	birthday
曾华	男	1977-09-01 00:00:00
匡明	男	1975-10-02 00:00:00
王丽	女	1976-01-23 00:00:00
李军	男	1976-02-20 00:00:00
王芳	女	1975-02-10 00:00:00
陆军	男	1974-06-03 00:00:00
王尼玛	男	1976-02-20 00:00:00
张全蛋	男	1975-02-10 00:00:00
赵铁柱	男	1974-06-03 00:00:00
张飞	男	1974-06-03 00:00:00
李诚	男	1958-12-02 00:00:00
王萍	女	1972-05-05 00:00:00
刘冰	女	1977-08-14 00:00:00
张旭	男	1969-03-12 00:00:00

-- 30. 查询所有'女'教师和'女'学生的名字,sex,birthday

```
SELECT s_name AS name, s_sex AS sex, s_birthday AS birthday FROM student WHERE s_sex = '女'
UNION
SELECT t_name AS name, t_sex AS sex, t_birthday AS birthday FROM teacher WHERE t_sex = '女';
```

name	sex	birthday
王丽	女	1976-01-23 00:00:00
王芳	女	1975-02-10 00:00:00
王萍	女	1972-05-05 00:00:00
刘冰	女	1977-08-14 00:00:00

-- 31. 查询成绩比该课程平均成绩低的同学的成绩表

注意:我的数据和视频中的数据有点不一样,所以查询结果有点区别
视频中score:

s_no	c_no	sc_degree
101	3-105	90
102	3-105	91
103	3-105	92
103	3-245	86
103	6-166	85
104	3-105	89
105	3-105	88
105	3-245	75
105	6-166	79
109	3-105	76
109	3-245	68
109	6-166	81

我数据库中: 109 | 6-166 | 81

```
SELECT * FROM score AS sc1 WHERE sc1.sc_degree < (SELECT AVG(sc_degree) FROM score AS sc2 WHERE
sc1.c_no = sc2.c_no);
```

s_no	c_no	sc_degree
105	3-245	75
105	6-166	79
109	3-105	76
109	3-245	68
109	6-166	81

```
+-----+-----+-----+-----+
```

进阶,显示学生name,课程name以及分数

```
SELECT s.s_name ,sc1.c_no,c.c_name, sc1.sc_degree FROM score AS sc1,student AS s,course AS c WHERE
sc1.sc_degree < (SELECT AVG(sc_degree) FROM score AS sc2 WHERE sc1.c_no = sc2.c_no) AND sc1.s_no =
s.s_no AND sc1.c_no = c.c_no ;
```

```
+-----+-----+-----+-----+
| s_name | c_no | c_name | sc_degree |
+-----+-----+-----+
| 赵铁柱 | 3-105 | 计算机导论 | 76 |
| 王芳 | 3-245 | 操作系统 | 75 |
| 赵铁柱 | 3-245 | 操作系统 | 68 |
| 王芳 | 6-166 | 数字电路 | 79 |
| 赵铁柱 | 6-166 | 数字电路 | 81 |
+-----+-----+-----+-----+
```

-- 32.查询所有任课教师的t_name 和 t_depart(要在分数表中可以查得到)

```
SELECT * FROM teacher WHERE t_no IN(SELECT t_no FROM course);
```

```
+-----+-----+-----+-----+-----+-----+
| t_no | t_name | t_sex | t_birthday | t_rof | t_depart |
+-----+-----+-----+-----+-----+-----+
| 804 | 李诚 | 男 | 1958-12-02 00:00:00 | 副教授 | 计算机系 |
| 825 | 王萍 | 女 | 1972-05-05 00:00:00 | 助教 | 计算机系 |
| 831 | 刘冰 | 女 | 1977-08-14 00:00:00 | 助教 | 电子工程系 |
| 856 | 张旭 | 男 | 1969-03-12 00:00:00 | 讲师 | 电子工程系 |
+-----+-----+-----+-----+-----+-----+
```

注意:我个人是从score表中查出有过考试成绩的课程,再用该课程查出教师的,因为当时我个人认为只有有考试成绩才算"任课",既然我写出来了,那我就放出来:(根据具体业务)

```
SELECT * FROM teacher WHERE t_no IN(SELECT t_no FROM course WHERE c_no IN (SELECT c_no from score));
```

```
+-----+-----+-----+-----+-----+-----+
| t_no | t_name | t_sex | t_birthday | t_rof | t_depart |
+-----+-----+-----+-----+-----+-----+
| 804 | 李诚 | 男 | 1958-12-02 00:00:00 | 副教授 | 计算机系 |
| 825 | 王萍 | 女 | 1972-05-05 00:00:00 | 助教 | 计算机系 |
| 856 | 张旭 | 男 | 1969-03-12 00:00:00 | 讲师 | 电子工程系 |
+-----+-----+-----+-----+-----+-----+
```

-- 33.查出至少有2名男生的班号

```
SELECT s_class FROM student WHERE s_sex = '男' GROUP BY s_class HAVING COUNT(s_no) > 1;
```

```
+-----+
| s_class |
+-----+
| 95033 |
| 95031 |
+-----+
```

-- 34.查询student 表中 不姓"王"的同学的记录

```
SELECT * FROM student WHERE s_name NOT LIKE '王%';
```

```
+-----+-----+-----+-----+-----+
| s_no | s_name | s_sex | s_birthday | s_class |
+-----+-----+-----+-----+-----+
| 101 | 曾华 | 男 | 1977-09-01 00:00:00 | 95033 |
| 102 | 匡明 | 男 | 1975-10-02 00:00:00 | 95031 |
| 104 | 李军 | 男 | 1976-02-20 00:00:00 | 95033 |
| 106 | 陆军 | 男 | 1974-06-03 00:00:00 | 95031 |
| 108 | 张全蛋 | 男 | 1975-02-10 00:00:00 | 95031 |
| 109 | 赵铁柱 | 男 | 1974-06-03 00:00:00 | 95031 |
| 110 | 张飞 | 男 | 1974-06-03 00:00:00 | 95038 |
+-----+-----+-----+-----+-----+
```

-- 35. 查询student 中每个学生的姓名和年龄(当前时间 - 出生年份)

```
SELECT s_name, YEAR(NOW()) - YEAR(s_birthday) AS age FROM student;
```

s_name	age
曾华	42
匡明	44
王丽	43
李军	43
王芳	44
陆军	45
王尼玛	43
张全蛋	44
赵铁柱	45
张飞	45

-- 36. 查询student中最大和最小的 s_birthday的值

```
SELECT MAX(s_birthday),MIN(s_birthday) FROM student;
```

MAX(s_birthday)	MIN(s_birthday)
1977-09-01 00:00:00	1974-06-03 00:00:00

-- 37.以班级号和年龄从大到小的顺序查询student表中的全部记录

```
SELECT * FROM student ORDER BY s_class DESC, s_birthday;
```

s_no	s_name	s_sex	s_birthday	s_class
110	张飞	男	1974-06-03 00:00:00	95038
103	王丽	女	1976-01-23 00:00:00	95033
104	李军	男	1976-02-20 00:00:00	95033
107	王尼玛	男	1976-02-20 00:00:00	95033
101	曾华	男	1977-09-01 00:00:00	95033
106	陆军	男	1974-06-03 00:00:00	95031
109	赵铁柱	男	1974-06-03 00:00:00	95031
105	王芳	女	1975-02-10 00:00:00	95031
108	张全蛋	男	1975-02-10 00:00:00	95031
102	匡明	男	1975-10-02 00:00:00	95031

-- 38.查询"男"教师 及其所上的课

```
SELECT * FROM course WHERE t_no IN (SELECT t_no FROM teacher WHERE t_sex = '男');
```

c_no	c_name	t_no
3-245	操作系统	804
6-166	数字电路	856

进阶:查出教师名称,教师性别, 课程名字

```
SELECT t.t_name,t.t_sex ,c.c_name FROM teacher t ,course c WHERE t_sex = '男' AND t.t_no = c.t_no;
```

t_name	t_sex	c_name
李诚	男	操作系统
张旭	男	数字电路

-- 39.查询最高分同学的s_no c_no 和 sc_degree;

```
SELECT * FROM score WHERE sc_degree = (select MAX(sc_degree) AS sc_degree FROM score);
```

s_no	c_no	sc_degree
103	3-105	92

-- 40. 查询和"李军"同性别的所有同学的s_name

```
SELECT s_name, s_sex FROM student WHERE s_SEX = (SELECT s_sex FROM student WHERE s_name = '李军');
```

s_name	s_sex
曾华	男
匡明	男
李军	男
陆军	男
王尼玛	男
张全蛋	男
赵铁柱	男
张飞	男

-- 41. 查询和"李军"同性别并且同班的所有同学的s_name

```
SELECT s_name, s_sex FROM student WHERE s_sex = (SELECT s_sex FROM student WHERE s_name = '李军') AND s_class = (SELECT s_class FROM student WHERE s_name = '李军');
```

s_name	s_sex
曾华	男
李军	男
王尼玛	男

```
SELECT s_name, s_sex FROM student s1 WHERE s_sex = (SELECT s_sex FROM student s2 WHERE s_name = '李军' AND s1.s_class = s2.s_class);
```

s_name	s_sex
曾华	男
李军	男
王尼玛	男

-- 42. 查询所有选修'计算机导论'课程的'男'同学的成绩表

```
SELECT * FROM score WHERE c_no = (SELECT c_no FROM course WHERE c_name = '计算机导论') AND s_no IN(SELECT s_no FROM student WHERE s_sex = '男');
```

s_no	c_no	sc_degree
101	3-105	90
102	3-105	91
104	3-105	89
109	3-105	76

进阶: 显示出s_name, c_name 替代之前的 s_no and c_no

```
SELECT s_name, c_name, sc_degree FROM score, student, course WHERE score.c_no = (SELECT c_no FROM course WHERE c_name = '计算机导论') AND score.s_no IN(SELECT s_no FROM student WHERE s_sex = '男') AND score.s_no = student.s_no AND score.c_no = course.c_no ;
```

s_name	c_name	sc_degree
曾华	计算机导论	90
匡明	计算机导论	91
李军	计算机导论	89

-- 43. 假设使用了以下命令建立了一个grade表

```
CREATE TABLE grade(  
    low INT(3),  
    upp INT(3),  
    grade CHAR(1)  
);  
INSERT INTO grade VALUES(90,100,'A');  
INSERT INTO grade VALUES(80,89,'B');  
INSERT INTO grade VALUES(70,79,'c');  
INSERT INTO grade VALUES(60,69,'D');  
INSERT INTO grade VALUES(0,59,'E');
```

-- 查询所有同学的s_no , c_no 和grade列

```
SELECT s_no, c_no , grade FROM score, grade WHERE sc_degree BETWEEN low and upp;
```

s_no	c_no	grade
101	3-105	A
102	3-105	A
103	3-105	A
103	3-245	B
103	6-166	B
104	3-105	B
105	3-105	B
105	3-245	c
105	6-166	c
109	3-105	c
109	3-245	D
109	6-166	B

进阶:显示学生名字和课程名称

```
SELECT s.s_no, s.s_name, c.c_name ,c.c_no , grade FROM student s, course c ,score sc, grade WHERE  
sc_degree BETWEEN low and upp AND sc.s_no = s.s_no AND sc.c_no = c.c_no;
```

s_no	s_name	c_name	c_no	grade
101	曾华	计算机导论	3-105	A
102	匡明	计算机导论	3-105	A
103	王丽	计算机导论	3-105	A
103	王丽	操作系统	3-245	B
103	王丽	数字电路	6-166	B
104	李军	计算机导论	3-105	B
105	王芳	计算机导论	3-105	B
105	王芳	操作系统	3-245	c
105	王芳	数字电路	6-166	c
109	赵铁柱	计算机导论	3-105	c
109	赵铁柱	操作系统	3-245	D
109	赵铁柱	数字电路	6-166	B