```sql
ALTER TABLE `course`
 DROP FOREIGN KEY `fk_course`;
ALTER TABLE `score`
 DROP FOREIGN KEY `fk_score`;
ALTER TABLE `score`
 DROP FOREIGN KEY `fk_score_1`;

DROP TABLE `student`;
DROP TABLE `course`;
 DROP TABLE `score`;
 DROP TABLE `teacher`;

 CREATE TABLE `student`
 (
  `id` varchar(20) NOT NULL COMMENT '编号',
  `name` varchar(20) NOT NULL COMMENT '姓名',
  `gender` varchar(20) NOT NULL COMMENT '性别',
  `birthday` datetime NULL COMMENT '出生年月',
  `class` varchar(20) NULL COMMENT '所在班级',
  PRIMARY KEY (`id`)
 )
  COMMENT = '学生表';
```

```sql
CREATE TABLE `course`
(
 `id` varchar(20) NOT NULL COMMENT '编号',
 `name` varchar(20) NOT NULL COMMENT '名称',
 `teacher_id` varchar(20) NOT NULL COMMENT '老师编号',
 PRIMARY KEY (`id`)
)
 COMMENT = '课程表';
```

```sql
CREATE TABLE `score`
(
 `student_id` varchar(20) NOT NULL COMMENT '学号',
 `course_id` varchar(20) NOT NULL COMMENT '课程编号',
 `degree` decimal(4, 1) NULL COMMENT '成绩',
 PRIMARY KEY (`student_id`, `course_id`)
)
```

```
 COMMENT = '成绩表';
```

```
CREATE TABLE `teacher`
(
 `id` varchar(20) NOT NULL COMMENT '编号',
 `name` varchar(20) NOT NULL COMMENT '姓名',
 `gender` varchar(20) NOT NULL COMMENT '姓名',
 `birthday` datetime NULL COMMENT '出生年月',
 `title` varchar(20) NULL COMMENT '职称',
 `department` varchar(20) NOT NULL COMMENT '部门',
 PRIMARY KEY (`id`)
)
 COMMENT = '教师表';
```

```
ALTER TABLE `course`
 ADD CONSTRAINT `fk_course` FOREIGN KEY (`teacher_id`) REFERENCES `teache
r` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE `score`
 ADD CONSTRAINT `fk_score` FOREIGN KEY (`student_id`) REFERENCES `student
` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE `score`
 ADD CONSTRAINT `fk_score_1` FOREIGN KEY (`course_id`) REFERENCES `course
` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION;
```

```
# 添加学生信息
insert into student
values ('108', '曾华', '男', '1977-09-01', '95033');
insert into student
values ('105', '匡明', '男', '1975-10-02', '95031');
insert into student
values ('107', '王丽', '女', '1976-01-23', '95033');
insert into student
values ('101', '李军', '男', '1976-02-20', '95033');
 insert into student
 values ('109', '王芳', '女', '1975-02-10', '95031');
 insert into student
 values ('103', '陆君', '男', '1974-06-03', '95031');

```

```
# 添加教师表
```

```
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', '9-888', '64');
 insert into score
 values ('105', '9-888', '91');
 insert into score
 values ('109', '9-888', '78');
 insert into score
 values ('103', '6-166', '85');
```

```
insert into score
values ('105', '6-166', '79');
insert into score
values ('109', '6-166', '81');
```

```
# 表的关系:
# 一对一: 每个用户都有一个身份证, 每个身份证对应一个用户
# 用户表(id, 账号, 密码)
# 1, admin, admin123
# 2, admin123, admin321
# 身份证表(id, 姓名, 性别, 出生年月, 身份证号, 家庭地址)
# 1, 张三, 男, 2018年12月28日17:15:12, 12321312312312312, 河南郑州
# 2, 张三, 男, 2018年12月28日17:15:12, 12321312312312312, 河南郑州

# 一对多: 一把锁有多把钥匙, 每个钥匙对应一把锁

# 多对多: 一个供应商可以为多个商家服务, 每个商家有多个供应商
# 必须有中间表

# 供应商(id, name)
# 1, 蒙牛
# 2, 伊利
# 商家(id, name)
# 10, ##
# 11, ##
# 中间表(供应商_id, 商家_id, 产品数量)
# 1, 10, 100
# 2, 10, 200
# 1, 11, 50
# 2, 11, 150

```

练习开始--------

1、 查询教师所有的部门

```
select distinct department
from teacher;
```

2、 查询student表的所有记录

```sql
select *
from student;
```

3、 查询score表中成绩在60到80之间的所有记录。

```sql
select *
from score
where degree between 60 and 80;
```

```sql

select *
from score
where degree >= 60
 and degree <= 80;
```

4、 查询score表中成绩为85，86或88的记录。

```sql
select *
from score
where degree in (85, 86, 88);
```

```sql
select *
from score
where degree = 85
 or degree = 86
 or degree = 88;
```

5、 查询student表中“95031”班或性别为“女”的同学记录。

```sql
select *
from student
where class = '95031'
 or gender = '女';
```

6、 以class降序查询student表的所有记录。

```sql
select *
from student
order by class desc;
```

7、 以course_id升序、degree降序查询score表的所有记录。

```
select *
from score
order by course_id asc, degree desc;
```

```
select *
from score
order by course_id, degree desc;
```

8、 查询“95031”班的学生人数。

```
select class, count(*)
from student
where class = '95031';
```

9、 查询score表中的最高分的学生学号和课程号。

```
select student_id, course_id
from score
where degree = (select max(degree)
 from score);
```

10、 查询每门课的平均成绩(保留两位小数)。

```
select c.name, format(avg(s.degree), 2)
from score s
 join course c on s.course_id = c.id
group by s.course_id;
```

```
select c.name, format(avg(s.degree), 2)
from score s,
 course c
where s.course_id = c.id
group by s.course_id;
```

```
select (select c.name from course c where c.id = s.course_id),
format(avg(s.degree), 2)
```

```
from score s
group by s.course_id;
```

11、 查询score表中至少有5名学生选修的并以3开头的课程的平均分数。

```
select course_id, format(avg(degree), 2)
from score
group by course_id
having count(student_id) >= 5
 and course_id like '3%';
```

```
select course_id, format(avg(degree), 2)
from score
where course_id like '3%'
group by course_id
having count(student_id) >= 5;
```

13、 查询分数大于70，小于90的student_id。

```
select student_id
from score
where degree > 70
 and degree < 90;
```

14、 查询所有学生的姓名、课程号和成绩。

```
select s1.name, s.course_id, s.degree
from score s
 join student s1 on s.student_id = s1.id;
```

15、 查询所有学生的学号、课程名和成绩。

```
select s.student_id, c.name, s.degree
from score s
 join course c on s.course_id = c.id;
```

16、 查询所有学生的姓名、课程名和成绩。

```sql
select s1.name, c.name, s.degree
from score s
 join student s1 on s.student_id = s1.id
 join course c on s.course_id = c.id;
```

17、 查询“95033”班学生的平均分。

```sql
select format(avg(degree), 2)
from score s
 join student s1 on s.student_id = s1.id and s1.class = '95033';
```

```sql
select format(avg(degree), 2)
from score s
 join student s1 on s.student_id = s1.id
where s1.class = '95033';
```

```sql
select format(avg(degree), 2)
from score
where student_id in (select id from student where class = '95033');
```

18、 假设使用如下命令建立了一个grade表：

```sql
create table grade
(
 low int(3),
 upp int(3),
 level char(1)
);
insert into grade
values (90, 100, 'A'),
 (80, 89, 'B'),
  (70, 79, 'C'),
  (60, 69, 'D'),
  (0, 59, 'E');
```

现查询所有同学的学号、课程号和等级。

```sql
select s.student_id, s.course_id, s.degree, g.level
from score s
 join grade g on s.degree between g.low and g.upp;
```

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 degree > (select degree from score
where student_id = '109' and course_id = '3-105');
```

```
select s1.*
from score s1
 join score s2 on s2.student_id = '109' and
 s2.course_id = '3-105' and s1.degree > s2.degree;
```

20、查询和学号为108、101的同学同年出生的所有学生的学号、姓名和出生年月。

```
select id, name, birthday
from student
where year(birthday) in (select year(birthday) from student
where id in ('108', '101'));
```

```
select s1.id, s1.name, s1.birthday
from student s1
 join student s2 on s2.id in ('108', '101') and year(s1.birthday) =
year(s2.birthday);

```

21、查询“张旭“教师任课的学生成绩。

```
select s.student_id, s.degree
from score s
 join course c on s.course_id = c.id
 join teacher t on c.teacher_id = t.id and t.name = '张旭';
```

```
select student_id, degree
from score
where course_id in (select id from course where teacher_id
in (select id from teacher where name = '张旭'));
```

22、查询选修某课程的同学人数多于5人的教师姓名。

```
select t.name
from score s
 join course c on s.course_id = c.id
 join teacher t on c.teacher_id = t.id
group by s.course_id
having count(*) >= 5;
```

23、查询95033班和95031班全体学生的记录。

```
select *
from student
where class in ('95033', '95031');
```

24、查询存在有85分以上成绩的课程编号.

```
select distinct course_id
from score
where degree >= 85;
```

25、查询出“计算机系“教师所教课程的成绩表。

```
select s.*
from score s
 join course c on s.course_id = c.id
 join teacher t on c.teacher_id = t.id and t.department = '计算机系';
```

26、查询“计算机系”与“电子工程系“不同职称的教师的姓名和职称。

```
select *
from teacher
where department = '计算机系'
 and title not in (select distinct title from teacher where department = '电子工程系')
union all
select *
from teacher
where department = '电子工程系'
```

```
  and title not in (select distinct title from teacher where department = '计算机系');
```

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”的同学)的学号、课程好和分数, 并按degree从高到低次序排序。

```
select *
from score
where course_id = '3-105'
 and degree > any (select degree from score where course_id = '3-245')
order by degree desc;
```

```
select *
from score
where course_id = '3-105'
 and degree > (select min(degree) from score where course_id = '3-245')
order by degree desc;
```

```
select distinct s1.student_id, s1.course_id, s1.degree
from score s1
 join score s2 on s1.course_id = '3-105'
 and s2.course_id = '3-245' and s1.degree > s2.degree;
```

28、查询选修编号为“3-105”且成绩高于选修编号为“3-245”课程的同学的学号、课程号和分数

```
select *
from score
where course_id = '3-105'
 and degree > all (select degree from score where course_id = '3-245');
```

```
select *
from score
where course_id = '3-105'
 and degree > (select max(degree) from score where course_id = '3-245');
```

29、查询所有教师和同学的name、gender和birthday.

```sql
select name, gender, birthday
from teacher
union all
select name, gender, birthday
from student;
```

30、查询所有“女”教师和“女”同学的name、gender和birthday.

```sql
select name, gender, birthday
from teacher
where gender = '女'
union all
select name, gender, birthday
from student
where gender = '女';
```

31、查询成绩比该课程平均成绩低的同学的成绩表。

```sql
select *
from score s1
where degree < (select avg(degree) from score s2
where s1.course_id = s2.course_id);

```

```sql
select s.*
from score s,
 (select course_id, avg(degree) a from score group by course_id) t
where s.course_id = t.course_id
 and s.degree < t.a;
```

32、查询所有任课教师的姓名和部门

```sql
select name, department
from teacher
where id in (select teacher_id from course);
```

33、查询所有未讲课的教师的姓名和部门.

```sql
select name, department
```

```
from teacher
where id not in (select teacher_id from course);
```

34、查询至少有2名男生的班号。

```
select class
from student
where gender = '男'
group by class
having count(*) >= 2;
```

35、查询student表中不姓“王”的同学记录。

```
select *
from student
where name not like '王%';
```

36、查询student表中每个学生的姓名和年龄。

```
select name, year(now()) - year(birthday) age
from student;

```

37、查询student表中最大和最小的birthday日期值。

```
select max(birthday) `最小`, min(birthday) `最大`
from student;
```

38、以班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 class desc, birthday asc;
```

39、查询“男”教师及其所上的课程。

```
select *
from teacher t
 left join course c on t.id = c.teacher_id
where t.gender = '男';
```

40、查询最高分同学的学生编号、课程编号和成绩。

```sql
select *
from score
where degree = (select max(degree) from score);
```

41、查询和“李军”同性别的所有同学的name.

```sql
select *
from student
where gender in (select distinct gender from student where name = '李军');
```

42、查询和“李军”同性别并同班的同学name.

```sql
select *
from student
where (gender, class) in (select gender, class from student where name = '李军') and name != '李军';
```

43、查询所有选修“计算机导论”课程的“男”同学的成绩表。

```sql
select st.name, s.*
from score s
 join course c on s.course_id = c.id and c.name = '计算机导论'
 join student st on s.student_id = st.id and st.gender = '男'
```