mysql测试题

表结构

请创建如下表，并创建相关约束

班级表：class

cid	caption
1	三年二班
2	一年三班
3	三年一班

学生表：student

sid	sname	gender	class_id
1	钢蛋	女	1
2	铁锤	女	1
3	山炮	男	2

老师表：teacher

tid	tname
1	波多
2	苍空
3	饭岛

课程表：course

cid	cname	tearch_id
1	生物	1
2	体育	1
3	物理	2

成绩表：score

sid	student_id	corse_id	number
1	1	1	60
2	1	2	59
3	2	2	100

创建表

```
create table class(
   cid int(11) not null auto_increment,
   caption varchar(32) not null,
   primary key (cid)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8;

create table teacher(
   tid int(11) not null auto_increment,
   tname varchar(32) not null,
   primary key (tid)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;

create table course(
   cid int(11) not null auto_increment,
   cname varchar(32) not null,
   teacher_id int(11) not null,
   primary key (cid),
   constraint fk_course_teacher foreign key(teacher_id) references teacher(tid)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8;

create table student(
```

```
    sid int(11) not null auto_increment,
    gender char(1) not null,
    class_id int(11) not null,
    sname varchar(32) not null,
    primary key (sid),
    constraint fk_class foreign key(class_id) references class(cid)
) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=utf8;

create table score(
    sid int(11) not null auto_increment,
    student_id int(11) not null,
    course_id int(11) not null,
    num int(11) not null,
    primary key (sid),
    constraint fk_score_course   foreign key (course_id) references  course(cid),
    constraint fk_score_student  foreign key (student_id) references student (sid)
) ENGINE=InnoDB AUTO_INCREMENT=53 DEFAULT CHARSET=utf8;
```

1.查询“生物”课程比“物理”课程成绩高的所有学生的学号.

```
#查询出生物课程对应的学生的学号 和 成绩
(select student_id, num as sw_score from score left join course on score.course_id =
course.cid where cname = "生物") as A
#同理查询出物理课程对应的学生的学号 和 成绩
(select student_id, num as wl_score from score left join course on score.course_id =
course.cid where cname = "物理") as B

#将上述结果连表进行查询
select A.student_id, sw_score,wl_score from
(select student_id, num as sw_score from score left join course on score.course_id =
course.cid where cname = "生物") as A
left join
(select student_id, num as wl_score from score left join course on score.course_id =
course.cid where cname = "物理") as B
on A.student_id = B.student_id where sw_score > wl_score;
```

2. 查询平均成绩大于60分的同学的学号 id 和平均成绩

```
select student_id,avg(num) from score group by student_id having avg(num)>60;
```

3. 查询所有同学的学号 id 、姓名、选课数、总成绩;

```
# 先从score表中根据student_id进行分组,得到临时表A,然后A表与学生表进行连接查询结果
(select student_id,count(course_id) as course_count,sum(num) as total from score group by
student_id)as A;

select student_id,sname,course_count,total from (select student_id,count(course_id) as
course_count,sum(num) as total from score group by student_id)as A left join student on
A.student_id = student.sid;
```

4. 查询姓“李”的老师的个数

```
select count(1) from teacher where tname like '李%';
```

5. 查询没学过“李平”老师课的同学的学号、姓名

```
#先查到李平老师所有课的id
select cid from course left join teacher on course.teacher_id  = teacher.tid where tname =
"李平老师";

#再从score表中查出选过该课程id的学生id ,此处需要用到去重,因为可能有学生两个课都选到过
select distinct student_id from score where course_id in (select cid from course left join
teacher on course.teacher_id  = teacher.tid where tname = "李平老师");

#从student表中筛选出 不在上述结果中的学生信息
select * from student where sid not in (
select distinct student_id from score where course_id in (select cid from course left join
teacher on course.teacher_id  = teacher.tid where tname = "李平老师")
);
```

6.查询学过编号“001”的课程并且也学过编号“002”课程的同学的学号、姓名

```
#先从score表中查到 course_id = 1 和 course_id = 2 对应的学生id
select student_id, course_id from score where course_id = 1 or course_id = 2 ;

#将上述得到的虚拟表与学生表左连接,根据student_id进行分组,然后筛选出人数大于的学生id,和学生姓名
select student_id,sname from (select student_id, course_id from score where course_id = 1
or course_id = 2) as B left join student on B.student_id = student.sid group by student_id
having count(student_id)>1;
```

7.查询学过“李平”老师所教的所有课的同学的学号、姓名

```
#查询出李平老师教的所有课程id
select cid from course left join teacher on teacher_id = teacher.tid where tname = "李平老
师";
#从查询出学过李平老师课的学生的id
select student_id from score where course_id  in (select cid from course left join teacher
on teacher_id = teacher.tid where tname = "李平老师");

#将上述得到的虚拟表与学生表左连接,根据student_id进行分组,然后筛选出人数大于的学生id,和学生姓名
select student_id, sname from (select student_id from score where course_id  in (select cid
from course left join teacher on teacher_id = teacher.tid where tname = "李平老师")) as A
left join student on A.student_id = student.sid group by student_id having
count(student_id)>1;
```

8.查询课程编号“2”的成绩比课程编号“1”课程低的所有同学的学号、姓名;

```
#第一种方法

#查询出课程编号是2的学生的id和成绩
```

```
select student_id,num as 2_score from score where course_id = 2;
#查询出课程编号是1的学生的id和成绩
select student_id,num as 1_score from score where course_id = 1;
#将上面得到的结果虚拟表左连接筛选出学习课程1和课程2的学生id
select A.student_id from
(select student_id,num as 2_score from score where course_id = 2) as A
left join
(select student_id,num as 1_score from score where course_id = 1) as B
on A.student_id = B.student_id where 2_score > 1_score;
#查询出学生id 对应的学生姓名
select sid,sname from student where sid in
(select A.student_id from (select student_id,num as 2_score from score where course_id = 2)
as A
left join
(select student_id,num as 1_score from score where course_id = 1) as B
on A.student_id = B.student_id where 2_score > 1_score);
#--------------------------------------------------------------------------------
或者:

select student.sid ,student.sname from student
right join
(select A.student_id from (select student_id,num as 2_score from score where course_id = 2)
as A
left join
(select student_id,num as 1_score from score where course_id = 1) as B
on A.student_id = B.student_id where 2_score > 1_score) as C
on student.sid = C.student_id;

#-----------------------------------------------------------------------------------------
第二种方法

#查询出课程编号是3 的学生的id,姓名和成绩
select student.sid,student.sname,score.num as 2_score from student left join score on
student.sid = score.student_id where course_id = 2;

#查询出课程编号是3 的学生的id,姓名和成绩
select student.sid,student.sname,score.num as 1_score from student left join score on
student.sid = score.student_id where course_id = 1;

#查询出id = 2课程成绩大于id= 1 的课程成绩的学生id 和姓名

select A.sid,A.sname from
(select student.sid,student.sname,score.num as 2_score from student left join score on
student.sid = score.student_id where course_id = 2) as A
left join
(select student.sid,student.sname,score.num as 1_score from student left join score on
student.sid = score.student_id where course_id = 1) as B
on A.sid = B.sid where 2_score > 1_score;
```

9.查询有课程成绩小于60分的同学的学号、姓名

```
查询出课程成绩小于60分的学生的id
select distinct student_id from score where num<60;
#查询出对应学生的姓名
select sid ,sname from student where sid in(select distinct student_id from score where
num<60);
```

10.查询没有学全所有课的同学的学号、姓名

```
#查询出总的课程数量
select count(1) from course;
#先在score表中根据学生进行分组,查询出学完所有课程的学生的id
select student_id  from score group by student_id having count(1) = (select count(1) from
course);
#查询出学生的id和姓名
select sid,sname from student where sid not in (select student_id from score group by
student_id having count(1) = (select count(1) from course));
```

11.查询至少有一门课与学号为“1”的同学所学相同的同学的学号和姓名

```
#查询出学号为1的同学所学的所有课程
select course_id from score where student_id = 1;

#查询出所学课程在上述查询结果中的学生的id
select distinct student_id from score where course_id in(select course_id from score where
student_id = 1);

#查询学生的姓名和ID
select sid ,sname from student
right join
(select distinct student_id from score where course_id in(select course_id from score where
student_id = 1)) as A
on student.sid = A.student_id where sid !=1;
```

12.查询至少学过学号为“1”同学所有课的其他同学学号和姓名

```
#查询出编号为1的同学学过的所有课程
select course_id from score  where student_id = 1

#查询出
```