

<<<<<<< HEAD

# mysql考察

=====

mysql考察

## 建表

```
# 创建新的数据库test
create database test;
# 使用数据库test
use test;
#创建学生表
CREATE TABLE `Student`(
    `s_id` VARCHAR(20) comment'学生号',
    `s_name` VARCHAR(20) NOT NULL DEFAULT '' comment'学生姓名',
    `s_birth` VARCHAR(20) NOT NULL DEFAULT '' comment'学生生日',
    `s_sex` VARCHAR(10) NOT NULL DEFAULT '' comment'学生性别',
    PRIMARY KEY(`s_id`)
)engine = InnoDB default charset=utf8 comment'学生表';
# 创建课程表
CREATE TABLE `Course`(
    `c_id` VARCHAR(20) comment'课程ID',
    `c_name` VARCHAR(20) NOT NULL DEFAULT '' comment'课程名称',
    `t_id` VARCHAR(20) NOT NULL comment'老师ID',
    PRIMARY KEY(`c_id`)
)engine = InnoDB default charset=utf8 comment'课程表';
# 教师表
CREATE TABLE `Teacher`(
    `t_id` VARCHAR(20) comment'老师ID',
    `t_name` VARCHAR(20) NOT NULL DEFAULT '' comment'教师姓名',
    PRIMARY KEY(`t_id`)
)engine = InnoDB default charset=utf8 comment'教师表';
# 成绩表
CREATE TABLE `Score`(
    `s_id` VARCHAR(20) comment'学生ID',
    `c_id` VARCHAR(20) comment'课程ID',
    `s_score` INT(3) comment'课程分数',
    PRIMARY KEY(`s_id`,`c_id`)
)engine = InnoDB default charset=utf8 comment'成绩表';
# 插入学生表测试数据
insert into
    Student
values('01' , '赵雷' , '1990-01-01' , '男'),
      ('02' , '钱电' , '1990-12-21' , '男'),
      ('03' , '孙风' , '1990-05-20' , '男'),
      ('04' , '李云' , '1990-08-06' , '男'),
      ('05' , '周梅' , '1991-12-01' , '女'),
      ('06' , '吴兰' , '1992-03-01' , '女'),
      ('07' , '郑竹' , '1989-07-01' , '女'),
      ('08' , '王菊' , '1990-01-20' , '女');
# 课程表测试数据
insert into
```

```

Course
values
    ('01' , '语文' , '02'),
    ('02' , '数学' , '01'),
    ('03' , '英语' , '03');

# 教师表测试数据
insert into
    Teacher
values
    ('01' , '张三'),
    ('02' , '李四'),
    ('03' , '王五');

# 成绩表测试数据
insert into
    Score
values
    ('01' , '01' , 80),
    ('01' , '02' , 90),
    ('01' , '03' , 99),
    ('02' , '01' , 70),
    ('02' , '02' , 60),
    ('02' , '03' , 80),
    ('03' , '01' , 80),
    ('03' , '02' , 80),
    ('03' , '03' , 80),
    ('04' , '01' , 50),
    ('04' , '02' , 30),
    ('04' , '03' , 20),
    ('05' , '01' , 76),
    ('05' , '02' , 87),
    ('06' , '01' , 31),
    ('06' , '03' , 34),
    ('07' , '02' , 89),
    ('07' , '03' , 98);

```

## 第一题

题目1、查询"01"课程比"02"课程成绩高的学生的信息及课程分数

分析：

- 1、此题会用到的表有学生表(Student)、成绩表(Score)
- 2、将Score表进行自关联,分别查找课程号为01 和课程号为02 的学生id 创建临时表
- 3、对查找出来的结果通过where条件进行筛选，得到课程01成绩高于课程02的学生信息及分数

```

    t1.s_id          /*学生号*/
,t1.s_name          /*学生姓名*/
,t1.s_birth         /*学生生日*/
,t1.s_sex           /*学生性别*/
,t2.c_id            /*课程01*/
,t2.s_score         /*学生成绩*/
,t3.c_id            /*课程02*/
,t3.s_score         /*学生成绩*/
from
    Student t1      /*学生表*/
inner join
    Score t2        /*成绩表*/
on

```

```

t1.s_id = t2.s_id and t2.c_id =01
inner join
Score t3
/*成绩表*/
on
t1.s_id = t3.s_id and t3.c_id = 02
where
t2.s_score > t3.s_score;

```

结果

	s_id	s_name	s_birth	s_sex	t2.c_id	t2.s_score	t3.c_id	t3.s_score
1	02	钱电	1990-12-21	男	01	70	02	60
2	04	李云	1990-08-06	男	01	50	02	30

## 第二题

题目2、查询"01"课程比"02"课程成绩低的学生的信息及课程分数

分析：

- 1、此题会用到的表有学生表(Student)、成绩表(Score)
- 2、将Score表进行自关联,分别查找课程号为01 和课程号为02 的学生id 创建临时表
- 3、对查找出来的结果通过where条件进行筛选，得到课程01成绩高于课程02的学生信息及分数

```

select
    t1.s_id
    ,t1.s_name
    ,t1.s_birth
    ,t1.s_sex
    ,t2.c_id
    ,t2.s_score
    ,t3.c_id
    ,t3.s_score
from
    Student t1
inner join
    Score t2
on
    t1.s_id = t2.s_id and t2.c_id =01
inner join
    Score t3
/*成绩表*/
on
    t1.s_id = t3.s_id and t3.c_id = 02
where
    t2.s_score < t3.s_score;

```

结果

	s_id	s_name	s_birth	s_sex	t2.c_id	t2.s_score	t3.c_id	t3.s_score
1	01	赵雷	1990-01-01	男	01	80	02	90
2	05	周梅	1991-12-01	女	01	76	02	87

## 第三题

题目3、查询平均成绩大于等于60分的同学的学生编号和学生姓名和平均成绩

分析：

通过学生号将学生表与成绩表进行关联

使用学生id进行数据进行分组

使用having对平均成绩进行筛选

```
select
    t.s_id          /*学生编号*/
  ,t.s_name        /*学生姓名*/
  ,avg(t1.s_score) /*平均成绩*/
from
    student t      /*学生表*/
inner join
    score t1       /*成绩表*/
on
    t.s_id = t1.s_id
group by
    t.s_id
having
    avg(t1.s_score) > 60 ;
```

结果

	s_id	s_name	`avg(t1.s_score)`
1	01	赵雷	89.6667
2	02	钱电	70.0000
3	03	孙风	80.0000
4	05	周梅	81.5000
5	07	郑竹	93.5000

## 第四题

题目4、查询平均成绩小于60分的同学的学生编号和学生姓名和平均成绩（包括有成绩的和无成绩的）

分析：

通过学生号将学生表与成绩表进行关联

使用学生id进行数据进行分组

如果平均成绩为空则设其值为0

使用having对平均成绩进行筛选

```
select
    t.s_id          /*学生编号*/
  ,t.s_name        /*学生姓名*/
  /*,avg(t1.s_score)*/ /*平均成绩*/
  ,if(avg(t1.s_score) is null,0,avg(t1.s_score)) as c
from
    student t      /*学生表*/
left join
    score t1       /*成绩表*/
on
    t.s_id = t1.s_id
group by
    t.s_id
having
    if(avg(t1.s_score) is null,0,avg(t1.s_score)) < 60 ;
```

结果

	s_id	s_name	c
1	04	李云	33.3333
2	06	吴兰	32.5000
3	08	王菊	0.0000

解决在计算过程中遇到空值问题

`if(column_name is null,0,column_name)` 如果字段值为null则返回0, 如果不为0则返回该字段值  
内容

示例:

```
if(avg(t1.s_score) is null,0,avg(t1.s_score)) as c
```

## 第五题

题目5、查询所有同学的学生编号、学生姓名、选课总数、所有课程的总成绩

分析:

通过学生编号将学生表与成绩表进行关联

使用count()函数计算选课数目

使用sum()函数计算课程总成绩

通过学生编号进行分组

在此处需要注意空值问题, 同时需要注意所使用的关联方式是否能产生所需的结果

```
select
    t.s_id                      /*学生编号*/
  ,t.s_name                    /*学生姓名*/
  ,count(t1.c_id) score_nm     /*选课总数*/
  ,if(sum(t1.s_score) is null ,0,sum(t1.s_score)) sum_score          /*所有课程
    总成绩*/
from
    student t                  /*学生表*/
left join
    score t1                   /*成绩表*/
on
    t.s_id = t1.s_id
group by t.s_id;
```

结果

	s_id	s_name	score_nm	sum_score
1	01	赵雷	3	269
2	02	钱电	3	210
3	03	孙风	3	240
4	04	李云	3	100
5	05	周梅	2	163
6	06	吴兰	2	65
7	07	郑竹	2	187
8	08	王菊	0	0

## 第六题

## 题目6、查询"李"姓老师的数量

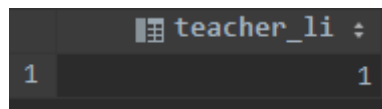
分析：

使用模糊查询查出“李”姓老师

使用count()函数统计“李”姓老师个数

```
select
    count(t_name) teacher_li    /*"李"姓老师个数*/
from
    teacher                    /*教师表*/
where
    t_name
like '李%';
```

结果



teacher_li
1

## 第七题

### 题目7、查询学过"张三"老师授课的同学的信息

分析：

通过学生编号将学生表与成绩表进行关联

通过课程编号将成绩表与课程表进行关联

通过教师编号将课程表与教师表进行关联

使用where筛选授课老师名为张三的老师

```
select
    t.s_id          /*学生id*/
    ,t.s_name       /*学生姓名*/
    ,t.s_birth      /*学生生日*/
    ,t.s_sex        /*学生性别*/
from
    student t       /*学生表*/
left join
    score t1        /*成绩表*/
on
    t.s_id = t1.s_id /*学生表的学生id与成绩表中的学生id*/
inner join
    course t2       /*课程表*/
on
    t1.c_id = t2.c_id
inner join
    teacher t3      /*教师表*/
on
    t2.t_id = t3.t_id
where
    t3.t_name = '张三';
```

结果

s_id	s_name	s_birth	s_sex
01	赵雷	1990-01-01	男
02	钱电	1990-12-21	男
03	孙风	1990-05-20	男
04	李云	1990-08-06	男
05	周梅	1991-12-01	女
07	郑竹	1989-07-01	女

## 第八题

题目8、查询没学过"张三"老师授课的同学的信息

分析：

学生表、老师表、授课表、成绩表

通过学生编号将学生表与成绩表进行关联

通过课程编号将成绩表与课程表进行关联

通过教师编号将课程表与教师表进行关联

使用学生id进行分组

使用having筛选授课老师不为张三的同学信息

本题由于使用了having语句，导致输出结果必须显示c\_id字段，有无优化方案

```
select
    t.s_id           /*学生id*/
  ,t.s_name         /*学生姓名*/
  ,t.s_birth        /*学生生日*/
  ,t.s_sex          /*学生性别*/
  ,t1.c_id
from
    student t           /*学生表*/
left join
    score t1           /*成绩表*/
on
    t.s_id = t1.s_id
left join
    course t2          /*课程表*/
on
    t1.c_id = t2.c_id
left join
    teacher t3         /*教师表*/
on
    t2.t_id = t3.t_id
group by
    t.s_id
having
    t.s_id not in
    (select s_id from score where c_id in
    (select c_id from course where t_id not in
    (select t_id from teacher where t_name != '张三'))
    group by s_id);
```

结果

	s_id	s_name	s_birth	s_sex	c_id
1	06	吴兰	1992-03-01	女	01
2	08	王菊	1990-01-20	女	<null>

## 第九题

题目9、查询学过编号为"01"并且也学过编号为"02"的课程的同学的信息

分析：

使用做左关联将学生表与成绩表进行关联

找出编号为"01"和"02"课程成绩不为空的学生id

通过学生id查询出对应的学生信息

在对同一字段进行筛选时，可采用自关联，并设定不同的条件

```
select
    t.*                                /*学过编号为"01"并且也学过编号为"02"的课程的学生信息
*/
from
    student t                          /*学生表*/
left join
    score t1                          /*成绩表*/
on
    t.s_id = t1.s_id and t1.c_id = 01
left join
    score t2                          /*成绩表*/
on
    t.s_id = t2.s_id and t2.c_id = 02
where
    t1.s_score is not null and t2.s_score is not null
group by
    t.s_id;
```

结果

	s_id	s_name	s_birth	s_sex
1	01	赵雷	1990-01-01	男
2	02	钱电	1990-12-21	男
3	03	孙风	1990-05-20	男
4	04	李云	1990-08-06	男
5	05	周梅	1991-12-01	女

## 第十题

题目10、查询学过编号为"01"但是没有学过编号为"02"的课程的同学的信息

分析：

使用做左关联将学生表与成绩表进行关联

找出编号为"01"课程成绩不为空且"02"课程成绩为空的学生id

通过学生id查询出对应的学生信息

```
select
    t.*                                /*学过编号为"01"但没学过编号为"02"的课程的学生信息
*/
from
```



```

student t                                /*学生表*/
left join
score t1                                /*成绩表*/
on
    t.s_id = t1.s_id and t1.c_id = 01
left join
score t2                                /*成绩表*/
on
    t.s_id = t2.s_id and t2.c_id = 02
where
    t1.s_score is not null and t2.s_score is null
group by
    t.s_id;

```

结果

	s_id	s_name	s_birth	s_sex
1	06	吴兰	1992-03-01	女

## 第十一题

题目11、查询没有学全所有课程的同学的信息

分析：

使用做左关联将学生表与成绩表进行关联

找出编号为"01","02","03"课程成绩任一门课程为空的学生id

通过学生id查询出对应的学生信息

```

select
    t.*                                /*学生信息*/
from
    student t                            /*学生表*/
left join
    score t1                            /*成绩表*/
on
    t.s_id = t1.s_id and t1.c_id = 01
left join
    score t2                            /*成绩表*/
on
    t.s_id = t2.s_id and t2.c_id = 02
left join
    score t3                            /*成绩表*/
on
    t.s_id = t3.s_id and t3.c_id = 03
where
    t1.s_score is null or t2.s_score is null or t3.s_score is null
group by
    t.s_id;

```

结果

	t.s_id	s_name	s_birth	s_sex	t1.s_id
1	01	赵雷	1990-01-01	男	01
2	02	钱电	1990-12-21	男	02
3	03	孙风	1990-05-20	男	03
4	04	李云	1990-08-06	男	04
5	05	周梅	1991-12-01	女	05
6	06	吴兰	1992-03-01	女	06
7	07	郑竹	1989-07-01	女	07
8	08	王菊	1990-01-20	女	<null>

## 第十二题

题目12、查询至少有一门课与学号为"01"的同学所学相同的同学的信息

分析：

查询学号id为01的学生的课程id

将学号为01的学生课程id与其他学生的课程id进行匹配，若有一门匹配成功则输出其个人信息

```
select
    t.*                                /*学生信息*/
from
    student t                        /*学生表*/
left join
    score t1                        /*成绩表*/
on
    t.s_id = t1.s_id
where
    t1.c_id
in (select c_id from score where s_id = 01 group by c_id) and t1.s_id != 01
group by
    s_id;
```

结果

	s_id	s_name	s_birth	s_sex
1	02	钱电	1990-12-21	男
2	03	孙风	1990-05-20	男
3	04	李云	1990-08-06	男
4	05	周梅	1991-12-01	女
5	06	吴兰	1992-03-01	女
6	07	郑竹	1989-07-01	女

## 第十三题

题目13、查询和"01"号的同学学习的课程完全相同的其他同学的信息

分析：

通过学生id将学生表与成绩表进行关联

通过取反筛选出课程与01号学生科目相同的课程

```
select
    t.*                                /*学生信息*/
from
    student t                        /*学生表*/
```

```

left join
    score t1                /*成绩表*/
on
    t.s_id = t1.s_id
where
    t.s_id
not in(select s_id from score where c_id
not in (select c_id from score where s_id=01))
and t.s_id != 01
group by
    t.s_id                /*学生id*/
having
    count(*) = (select count(*) from score where s_id = 01);

```

结果

	s_id	s_name	s_birth	s_sex
1	02	钱电	1990-12-21	男
2	03	孙风	1990-05-20	男
3	04	李云	1990-08-06	男

```

# 非正确写法（此写法仅统计课程数目与01号学生相同的学生信息）
select
    t.*                /*学生信息*/
    ,t1.s_id
from
    student t          /*学生表*/
inner join
    score t1           /*成绩表*/
on
    t.s_id = t1.s_id
group by
    t.s_id            /*学生学号*/
having
    count(t1.c_id) = (select count(score.c_id) from score where score.s_id = 01)
and t1.s_id != 01;

```

结果

	t.s_id	s_name	s_birth	s_sex	t1.s_id
1	02	钱电	1990-12-21	男	02
2	03	孙风	1990-05-20	男	03
3	04	李云	1990-08-06	男	04

## 第十四题

题目14、查询没学过"张三"老师讲授的任一门课程的学生姓名

分析：

通过教师id找出张三所对应的课程

根据课程id找出没有修张三课程的学生id

```

select
    t.s_name                /*学生信息*/
from

```

```

        student t                /*学生表*/
    left join
        score t1                 /*成绩表*/
    on
        t.s_id = t1.s_id
    left join
        course t2               /*课程表*/
    on
        t1.c_id = t2.c_id
    left join
        teacher t3              /*教师表*/
    on
        t2.t_id = t3.t_id
    group by
        t.s_id
    having
        t.s_id not in
        (select s_id from score where c_id in
        (select c_id from course where t_id not in
        (select t_id from teacher where t_name != '张三')))
    group by s_id);

```

结果

	s_name
1	吴兰
2	王菊

## 题目十五

题目15、查询两门及其以上不及格课程的同学的学号，姓名及其平均成绩

分析：

使用学生id进行分组

筛选学生成绩小于六十且科目数量大于2的学生信息

```

select
    t.s_id                /*学生编号*/
    ,t.s_name             /*学生姓名*/
    ,avg(t1.s_score)      /*平均成绩*/
    ,t1.s_score
from
    student t
left join
    score t1
on
    t.s_id = t1.s_id
group by
    t.s_id
having
    t1.s_score < 60 and count(t1.c_id) >= 2;

```

结果

	s_id	s_name	avg(t1.s_score)	s_score
1	04	李云	33.3333	50
2	06	吴兰	32.5000	31

## 第十六题

题目16、检索"01"课程分数小于60，按分数降序排列的学生信息

筛选出课程号为01 以及该课程分数小于60

根据课程分数进行降序排列

```
select
    t.*                                /*01课程分数小于60的学生信息*/
from
    student t                          /*学生表*/
inner join
    score t1                           /*成绩表*/
on
    t.s_id = t1.s_id
where
    t1.c_id = 01 and t1.s_score < 60
order by
    t1.s_score desc;
```

结果

	s_id	s_name	s_birth	s_sex
1	04	李云	1990-08-06	男
2	06	吴兰	1992-03-01	女

## 第十七题

题目17、按平均成绩从高到低显示所有学生的所有课程的成绩以及平均成绩

分析：

通过学生id将学生表与成绩表关联

使用学生id进行分组

根据学生的平均分从高到低输出学生成绩

```
# 学生平均成绩从高到低排序
select
    t.*                                /*学生信息*/
    ,avg(t1.s_score)                  /*平均成绩*/
from
    student t                          /*学生表*/
left join
    score t1                           /*成绩表*/
on
    t.s_id = t1.s_id
group by
    t1.s_id
order by
    avg(t1.s_score) desc;
```

结果

	s_id	s_name	s_birth	s_sex	if(round(avg(t1.s_score)...
1	07	郑竹	1989-07-01	女	93.50
2	01	赵雷	1990-01-01	男	89.67
3	05	周梅	1991-12-01	女	81.50
4	03	孙风	1990-05-20	男	80.00
5	02	钱电	1990-12-21	男	70.00
6	04	李云	1990-08-06	男	33.33
7	06	吴兰	1992-03-01	女	32.50
8	08	王菊	1990-01-20	女	0.00

```
# 依据学生平均成绩进行排序输出
select
    t.*,                                /*学生信息*/
    GROUP_CONCAT(t1.s_score) score      /*学生成绩*/
from
    student t                          /*学生表*/
left join
    score t1                          /*成绩表*/
on
    t.s_id = t1.s_id
group by
    t.s_id
order by
    avg(t1.s_score) desc;
```

结果

	s_id	s_name	s_birth	s_sex	score
1	07	郑竹	1989-07-01	女	89,98
2	01	赵雷	1990-01-01	男	80,90,99
3	05	周梅	1991-12-01	女	76,87
4	03	孙风	1990-05-20	男	80,80,80
5	02	钱电	1990-12-21	男	70,60,80
6	04	李云	1990-08-06	男	50,30,20
7	06	吴兰	1992-03-01	女	31,34
8	08	王菊	1990-01-20	女	<null>

## 第十八题

题目18、查询各科成绩最高分、最低分和平均分：以如下形式显示：课程ID，课程name，最高分，最低分，平均分，及格率，中等率，优良率，优秀率 及格为>=60，中等为：70-80，优良为：80-90，优秀为：>=90

分析：

通过课程id将成绩表与课程表进项关联

依据课程id进行分组

使用max()函数计算课程最高分

使用min()函数计算课程最低分

使用avg()函数计算课程平均分

通过case when语句进行条件筛选计算及格率、中等率、优良率、优秀率

此处计算中等率以及优良率采用的是左闭右开

```

select
    t.c_id                                /*课程id*/
    ,t1.c_name                            /*课程名称*/
    ,max(t.s_score) max                    /*最高分*/
    ,min(t.s_score) min                    /*最低分*/
    ,avg(t.s_score) avg                    /*平均分*/
    ,round(sum(case when t.s_score >= 60 then 1 else 0 end)/count(s_id),2) a1
                                          /*及格率*/
    ,round(sum(case when t.s_score >= 70 && t.s_score < 80 then 1 else 0
end)/count(s_id),2) a4                    /*中等率*/
    ,round(sum(case when t.s_score >= 80 && t.s_score < 90 then 1 else 0
end)/count(s_id),2) a3                    /*优良率*/
    ,round(sum(case when t.s_score > 90 then 1 else 0 end)/count(s_id),2) a2
                                          /*优秀率*/

from
    score t                                /*成绩表*/
left join
    course t1                             /*课程表*/
on
    t.c_id = t1.c_id
group by
    t.c_id;

```

结果

	c_id	c_name	max	min	avg	a1	a4	a3	a2
1	01	语文	80	31	64.5000	0.67	0.33	0.33	0.00
2	02	数学	90	30	72.6667	0.83	0.00	0.50	0.00
3	03	英语	99	20	68.5000	0.67	0.00	0.33	0.33

## 题目十九

题目19、按各科成绩进行排序，并显示排名

分析：

使用学生id将学生表与成绩表进行关联

首先根据可能id进行排序，再根据成绩进行降序排序

提仅根据课程id进行分组，但是在进行排序时是根据所有的成绩进行排序的

```

select
    t.*                                    /*学生姓名*/
    ,t1.*
    ,row_number() over ( order by s_score desc)
from
    student t                              /*学生表*/
left join
    score t1                              /*成绩表*/
on t.s_id = t1.s_id
left join
    score t2                              /*成绩表*/
on t.s_id = t2.s_id
group by
    t.s_id,t1.c_id
order by
    t1.c_id,t1.s_score desc;

```

结果

	t.s_id	s_name	s_birth	s_sex	t1.s_id	c_id	s_score	row_number() over ( order by s_sc...
1	08	王莉	1990-01-20	女	<null>	<null>	<null>	19
2	01	赵雷	1990-01-01	男	01	01	80	6
3	03	孙风	1990-05-20	男	03	01	80	8
4	05	周梅	1991-12-01	女	05	01	76	11
5	02	钱电	1990-12-21	男	02	01	70	12
6	04	李云	1990-08-06	男	04	01	50	14
7	06	吴兰	1992-03-01	女	06	01	31	16
8	01	赵雷	1990-01-01	男	01	02	90	3
9	07	郑竹	1989-07-01	女	07	02	89	4
10	05	周梅	1991-12-01	女	05	02	87	5
11	03	孙风	1990-05-20	男	03	02	80	9
12	02	钱电	1990-12-21	男	02	02	60	13
13	04	李云	1990-08-06	男	04	02	30	17
14	01	赵雷	1990-01-01	男	01	03	99	1
15	07	郑竹	1989-07-01	女	07	03	98	2
16	02	钱电	1990-12-21	男	02	03	80	7
17	03	孙风	1990-05-20	男	03	03	80	10
18	06	吴兰	1992-03-01	女	06	03	34	15
19	04	李云	1990-08-06	男	04	03	20	18

## 第二十题

题目20、查询学生的总成绩并进行排名

使用学生编号对学生表已经成绩表进行关联

通过学生id进行分组并使用sum函数计算学生成绩使用降序排列

```
select
    t.s_name                /*学生姓名*/
    ,sum(t1.s_score)        /*总成绩*/
from
    student t              /*学生表*/
inner join
    score t1               /*成绩表*/
on t.s_id = t1.s_id
group by
    t.s_id
order by
    sum(t1.s_score) desc;
```

结果

	s_name	sum(t1.s_score)
1	赵雷	269
2	孙风	240
3	钱电	210
4	郑竹	187
5	周梅	163
6	李云	100
7	吴兰	65

## 第二十一题

题目21、写一个函数，checkScore(s\_id) 返回该s\_id的分数最高的科目对应的教师ID。

分析：

写出查询语句

根据查询语句创建函数

调用函数

```
DELIMITER $                /*定义结束符为$*/
```



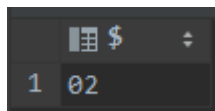
```

create function checkScore(sid varchar(10)) /*定义函数名及其传入的参数名*/
returns varchar(10) /*设置返回值类型*/
begin
return (
    select
        t2.t_id /*教师id*/
    from
        course t /*课程表*/
    inner join
        score t1 /*成绩表*/
    on
        t.c_id = t1.c_id and t1.s_id = sid
    inner join
        teacher t2 /*教师表*/
    on
        t.t_id = t2.t_id
    order by
        t1.s_score desc limit 1
);

end$
/*调用函数*/
select checkScore('04')$

```

结果



## 第二十二题

题目22、写一个过程，fixScore() 将每个s\_id分数最高的科目对应的教师ID存入一张表中。表自定义，字段需要有s\_id, s\_name, t\_name, c\_name

分析：

通过成绩表关联课程表、学生表，课程表关联教师表找出每个每个s\_id分数最高所对应的教师id

```

drop procedure fixScore;
create procedure fixScore() /*创建存储过程名称*/
begin
    select
        t.s_id /*学生id*/
        ,t2.s_name /*学生姓名*/
        ,t3.t_name /*老师姓名*/
        ,t3.t_id /*老师id*/
        ,t1.c_name /*课程名称*/
        ,max(t.s_score) max_score /*成绩*/
    from
        score t /*成绩表*/
    left join
        course t1 /*课程表*/
    on
        t.c_id = t1.c_id
    right join
        student t2 /*学生表*/
    on
        t.s_id = t2.s_id

```

```
left join
  teacher t3          /*教师表*/
on
  t1.t_id = t3.t_id
group by
  t.s_id;
end;
call fixScore;        /*调用存储过程*/
```

结果

	s_id	s_name	t_name	t_id	c_name	max_score
1	01	赵雷	李四	02	语文	99
2	02	钱电	李四	02	语文	80
3	03	孙风	李四	02	语文	80
4	04	李云	李四	02	语文	50
5	05	周梅	李四	02	语文	87
6	06	吴兰	李四	02	语文	34
7	07	郑竹	张三	01	数学	98
8	<null>	王菊	<null>	<null>	<null>	<null>

58e00a3 (测试)