

## 表名和字段

### -1.学生表

Student(s\_id,s\_name,s\_birth,s\_sex) --学生编号,学生姓名, 出生年月,学生性别

### -2.课程表

Course(c\_id,c\_name,t\_id) --课程编号, 课程名称, 教师编号

### -3.教师表

Teacher(t\_id,t\_name) --教师编号,教师姓名

### -4.成绩表

Score(s\_id,c\_id,s\_score) --学生编号,课程编号,分数

## 测试数据

--建表

```
--建表
--学生表
CREATE TABLE `Student` (
  `s_id` VARCHAR(20),
  `s_name` VARCHAR(20) NOT NULL DEFAULT '',
  `s_birth` VARCHAR(20) NOT NULL DEFAULT '',
  `s_sex` VARCHAR(10) NOT NULL DEFAULT '',
  PRIMARY KEY(`s_id`)
);
--课程表
CREATE TABLE `Course` (
  `c_id` VARCHAR(20),
  `c_name` VARCHAR(20) NOT NULL DEFAULT '',
  `t_id` VARCHAR(20) NOT NULL,
  PRIMARY KEY(`c_id`)
);
--教师表
CREATE TABLE `Teacher` (
  `t_id` VARCHAR(20),
  `t_name` VARCHAR(20) NOT NULL DEFAULT '',
  PRIMARY KEY(`t_id`)
);
--成绩表
CREATE TABLE `Score` (
  `s_id` VARCHAR(20),
  `c_id` VARCHAR(20),
  `s_score` INT(3),
  PRIMARY KEY(`s_id`,`c_id`)
);
--插入学生表测试数据
insert into Student values('01' , '赵雷' , '1990-01-01' , '男');
```

```

insert into Student values('02' , '钱电' , '1990-12-21' , '男');
insert into Student values('03' , '孙凤' , '1990-05-20' , '男');
insert into Student values('04' , '李云' , '1990-08-06' , '男');
insert into Student values('05' , '周梅' , '1991-12-01' , '女');
insert into Student values('06' , '吴兰' , '1992-03-01' , '女');
insert into Student values('07' , '郑竹' , '1989-07-01' , '女');
insert into Student values('08' , '王菊' , '1990-01-20' , '女');
--课程表测试数据
insert into Course values('01' , '语文' , '02');
insert into Course values('02' , '数学' , '01');
insert into Course values('03' , '英语' , '03');

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

```

```

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

```

## 练习题和sql语句

-- 1、查询"01"课程比"02"课程成绩高的学生的信息及课程分数

```

select a.* ,b.s_score as 01_score,c.s_score as 02_score from
student a
  join score b on a.s_id=b.s_id and b.c_id='01'
 left join score c on a.s_id=c.s_id and c.c_id='02' or c.c_id = NULL where
b.s_score>c.s_score

```

--也可以这样写

```

select a.*,b.s_score as 01_score,c.s_score as 02_score from student
a,score b,score c

```

```

where a.s_id=b.s_id
and a.s_id=c.s_id
and b.c_id='01'
and c.c_id='02'
and b.s_score>c.s_score

```

-- 2、查询"01"课程比"02"课程成绩低的学生信息及课程分数

```

select a.* ,b.s_score as 01_score,c.s_score as 02_score from
student a left join score b on a.s_id=b.s_id and b.c_id='01' or b.c_id=NULL
join score c on a.s_id=c.s_id and c.c_id='02' where b.s_score<c.s_score

```

-- 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

```

select b.s_id,b.s_name,ROUND(AVG(a.s_score),2) as avg_score from
student b
join score a on b.s_id = a.s_id
GROUP BY b.s_id,b.s_name HAVING avg_score >=60;

```

-- 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)

```

select b.s_id,b.s_name,ROUND(AVG(a.s_score),2) as avg_score from
student b
left join score a on b.s_id = a.s_id
GROUP BY b.s_id,b.s_name HAVING avg_score <60
union
select a.s_id,a.s_name,0 as avg_score from
student a
where a.s_id not in (
select distinct s_id from score);

```

-- 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

```

select a.s_id,a.s_name,count(b.c_id) as sum_course,sum(b.s_score) as sum_score
from
student a
left join score b on a.s_id=b.s_id
GROUP BY a.s_id,a.s_name;

```

-- 6、查询"李"姓老师的数量

```

select count(t_id) from teacher where t_name like '李%';

```

-- 7、查询学过"张三"老师授课的同学的信息

```

select a.* from
student a
join score b on a.s_id=b.s_id where b.c_id in(
select c_id from course where t_id =(
select t_id from teacher where t_name = '张三'));

```

-- 8、查询没学过"张三"老师授课的同学的信息

```

select * from
student c
where c.s_id not in(
select a.s_id from student a join score b on a.s_id=b.s_id where b.c_id
in(
select a.c_id from course a join teacher b on a.t_id = b.t_id where
t_name = '张三'));

```

-- 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

```

select a.* from
student a,score b,score c
where a.s_id = b.s_id and a.s_id = c.s_id and b.c_id='01' and c.c_id='02';

```

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

```
select a.* from
  student a
  where a.s_id in (select s_id from score where c_id='01' ) and a.s_id not
in(select s_id from score where c_id='02')
```

-- 11、查询没有学全所有课程的同学们的信息

--@wendiepei的写法

```
select s.* from student s
left join score s1 on s1.s_id=s.s_id
group by s.s_id having count(s1.c_id)<(select count(*) from course)
```

--@k1051785839的写法

```
select *
from student
where s_id not in(
select s_id from score t1
group by s_id having count(*) =(select count(distinct c_id) from course))
```

-- 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

```
select * from student where s_id in(
  select distinct a.s_id from score a where a.c_id in(select a.c_id from score a
where a.s_id='01')
);
```

-- 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

--@ouyang\_1993的写法

```
SELECT
  Student.*
FROM
  Student
WHERE
  s_id IN (SELECT s_id FROM Score GROUP BY s_id HAVING COUNT(s_id) = (
    #下面的语句是找到'01'同学学习的课程数
    SELECT COUNT(c_id) FROM Score WHERE s_id = '01'
  )
)
```

AND s\_id NOT IN (
 #下面的语句是找到学过'01'同学没学过的课程，有哪些同学。并排除他们

```
SELECT s_id FROM Score
WHERE c_id IN(
  #下面的语句是找到'01'同学没学过的课程
  SELECT DISTINCT c_id FROM Score
  WHERE c_id NOT IN (
    #下面的语句是找出'01'同学学习的课程
    SELECT c_id FROM Score WHERE s_id = '01'
  )
)
```

) GROUP BY s\_id
) #下面的条件是排除01同学

```
AND s_id NOT IN ('01')
```

--@k1051785839的写法

```
SELECT
  t3.*
FROM
  (
    SELECT
      s_id,
      group_concat(c_id ORDER BY c_id) group1
```

```

FROM
    score
WHERE
    s_id &lt;t;> '01'
GROUP BY
    s_id
) t1
INNER JOIN (
    SELECT
        group_concat(c_id ORDER BY c_id) group2
    FROM
        score
    WHERE
        s_id = '01'
    GROUP BY
        s_id
) t2 ON t1.group1 = t2.group2
INNER JOIN student t3 ON t1.s_id = t3.s_id

-- 14、查询没学过"张三"老师讲授的任一门课程的学生姓名
select a.s_name from student a where a.s_id not in (
    select s_id from score where c_id =
        (select c_id from course where t_id =(
            select t_id from teacher where t_name = '张三')));

-- 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
select a.s_id,a.s_name,ROUND(AVG(b.s_score)) from
student a
left join score b on a.s_id = b.s_id
where a.s_id in(
    select s_id from score where s_score<60 GROUP BY s_id having count(1)>=2)
GROUP BY a.s_id,a.s_name

-- 16、检索"01"课程分数小于60，按分数降序排列的学生信息
select a.*,b.c_id,b.s_score from
student a,score b
where a.s_id = b.s_id and b.c_id='01' and b.s_score<60 ORDER BY b.s_score
DESC;

-- 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
select a.s_id,(select s_score from score where s_id=a.s_id and c_id='01') as 语
文,
    (select s_score from score where s_id=a.s_id and c_id='02') as 数学,
    (select s_score from score where s_id=a.s_id and c_id='03') as 英语,
    round(avg(s_score),2) as 平均分 from score a GROUP BY a.s_id ORDER BY 平均分
DESC;
--@喝完这杯还有一箱的写法
SELECT a.s_id,MAX(CASE a.c_id WHEN '01' THEN a.s_score END ) 语文,
MAX(CASE a.c_id WHEN '02' THEN a.s_score END ) 数学,
MAX(CASE a.c_id WHEN '03' THEN a.s_score END ) 英语,
avg(a.s_score),b.s_name FROM Score a JOIN Student b ON a.s_id=b.s_id GROUP BY
a.s_id ORDER BY 5 DESC

-- 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
select a.c_id,b.c_name,MAX(s_score),MIN(s_score),ROUND(AVG(s_score),2),

```

```

ROUND(100*(SUM(case when a.s_score>=60 then 1 else 0 end)/SUM(case when
a.s_score then 1 else 0 end)),2) as 及格率,
ROUND(100*(SUM(case when a.s_score>=70 and a.s_score<=80 then 1 else 0
end)/SUM(case when a.s_score then 1 else 0 end)),2) as 中等率,
ROUND(100*(SUM(case when a.s_score>=80 and a.s_score<=90 then 1 else 0
end)/SUM(case when a.s_score then 1 else 0 end)),2) as 优良率,
ROUND(100*(SUM(case when a.s_score>=90 then 1 else 0 end)/SUM(case when
a.s_score then 1 else 0 end)),2) as 优秀率
from score a left join course b on a.c_id = b.c_id GROUP BY a.c_id,b.c_name

```

-- 19、按各科成绩进行排序，并显示排名

-- mysql没有rank函数

```

select a.s_id,a.c_id,
       @i:=@i +1 as i保留排名,
       @k:=(case when @score=a.s_score then @k else @i end) as rank不保留排名,
       @score:=a.s_score as score
from (
       select s_id,c_id,s_score from score GROUP BY s_id,c_id,s_score ORDER BY
s_score DESC
)a,(select @k:=0,@i:=0,@score:=0)s
--@k1051785839的写法
(select * from (select
t1.c_id,
t1.s_score,
(select count(distinct t2.s_score) from score t2 where t2.s_score>=t1.s_score
and t2.c_id='01') rank
FROM score t1 where t1.c_id='01'
order by t1.s_score desc) t1)
union
(select * from (select
t1.c_id,
t1.s_score,
(select count(distinct t2.s_score) from score t2 where t2.s_score>=t1.s_score
and t2.c_id='02') rank
FROM score t1 where t1.c_id='02'
order by t1.s_score desc) t2)
union
(select * from (select
t1.c_id,
t1.s_score,
(select count(distinct t2.s_score) from score t2 where t2.s_score>=t1.s_score
and t2.c_id='03') rank
FROM score t1 where t1.c_id='03'
order by t1.s_score desc) t3)

```

-- 20、查询学生的总成绩并进行排名

```

select a.s_id,
       @i:=@i+1 as i,
       @k:=(case when @score=a.sum_score then @k else @i end) as rank,
       @score:=a.sum_score as score
from (select s_id,SUM(s_score) as sum_score from score GROUP BY s_id ORDER BY
sum_score DESC)a,
(select @k:=0,@i:=0,@score:=0)s

```

-- 21、查询不同老师所教不同课程平均分从高到低显示

```

select a.t_id,c.t_name,a.c_id,ROUND(avg(s_score),2) as avg_score from course a
left join score b on a.c_id=b.c_id

```

```

left join teacher c on a.t_id=c.t_id
GROUP BY a.c_id,a.t_id,c.t_name ORDER BY avg_score DESC;

```

-- 22、查询所有课程的成绩第2名到第3名的学生信息及该课程成绩

```

select d.*,c.排名,c.s_score,c.c_id from (
    select a.s_id,a.s_score,a.c_id,@i:=@i+1 as 排名 from score a,
(select @i:=0)s where a.c_id='01'
    ORDER BY a.s_score DESC
)c
left join student d on c.s_id=d.s_id
where 排名 BETWEEN 2 AND 3
UNION
select d.*,c.排名,c.s_score,c.c_id from (
    select a.s_id,a.s_score,a.c_id,@j:=@j+1 as 排名 from score a,
(select @j:=0)s where a.c_id='02'
    ORDER BY a.s_score DESC
)c
left join student d on c.s_id=d.s_id
where 排名 BETWEEN 2 AND 3
UNION
select d.*,c.排名,c.s_score,c.c_id from (
    select a.s_id,a.s_score,a.c_id,@k:=@k+1 as 排名 from score a,
(select @k:=0)s where a.c_id='03'
    ORDER BY a.s_score DESC
)c
left join student d on c.s_id=d.s_id
where 排名 BETWEEN 2 AND 3;

```

-- 23、统计各科成绩各分数段人数：课程编号，课程名称，[100-85]，[85-70]，[70-60]，[0-60]及所占百分比

```

select distinct f.c_name,a.c_id,b.`85-100`,b.百分比,c.`70-85`,c.百分比,d.`60-70`,d.百分比,e.`0-60`,e.百分比 from score a
left join (select c_id,SUM(case when s_score >85 and s_score <=100 then
1 else 0 end) as `85-100`,
ROUND(100*(SUM(case when s_score >85 and s_score <=100
then 1 else 0 end)/count(*)),2) as 百分比
from score GROUP BY c_id)b on a.c_id=b.c_id
left join (select c_id,SUM(case when s_score >70 and s_score <=85 then 1
else 0 end) as `70-85`,
ROUND(100*(SUM(case when s_score >70 and s_score <=85 then
1 else 0 end)/count(*)),2) as 百分比
from score GROUP BY c_id)c on a.c_id=c.c_id
left join (select c_id,SUM(case when s_score >60 and s_score <=70 then 1
else 0 end) as `60-70`,
ROUND(100*(SUM(case when s_score >60 and s_score <=70 then
1 else 0 end)/count(*)),2) as 百分比
from score GROUP BY c_id)d on a.c_id=d.c_id
left join (select c_id,SUM(case when s_score >=0 and s_score <=60 then 1
else 0 end) as `0-60`,
ROUND(100*(SUM(case when s_score >=0 and s_score <=60 then
1 else 0 end)/count(*)),2) as 百分比
from score GROUP BY c_id)e on a.c_id=e.c_id
left join course f on a.c_id = f.c_id

```

-- 24、查询学生平均成绩及其名次

```

select a.s_id,
    @i:=@i+1 as '不保留空缺排名',
    @k:=(case when @avg_score=a.avg_s then @k else @i end) as '保留空缺排名',

```

```

        @avg_score:=avg_s as '平均分'
    from (select s_id,ROUND(AVG(s_score),2) as avg_s from score GROUP BY s_id
ORDER BY avg_s DESC)a,(select @avg_score:=0,@i:=0,@k:=0)b;
-- 25、查询各科成绩前三名的记录
    -- 1.选出b表比a表成绩大的所有组
    -- 2.选出比当前id成绩大的 小于三个的
select a.s_id,a.c_id,a.s_score from score a
left join score b on a.c_id = b.c_id and a.s_score<b.s_score
group by a.s_id,a.c_id,a.s_score HAVING COUNT(b.s_id)<3
ORDER BY a.c_id,a.s_score DESC

-- 26、查询每门课程被选修的学生数
select c_id,count(s_id) from score a GROUP BY c_id

-- 27、查询出只有两门课程的全部学生的学号和姓名
select s_id,s_name from student where s_id in(
    select s_id from score GROUP BY s_id HAVING COUNT(c_id)=2);

-- 28、查询男生、女生人数
select s_sex,COUNT(s_sex) as 人数 from student GROUP BY s_sex

-- 29、查询名字中含有"风"字的学生信息
select * from student where s_name like '%风%';

-- 30、查询同名同性学生名单，并统计同名人数
select a.s_name,a.s_sex,count(*) from student a JOIN
    student b on a.s_id !=b.s_id and a.s_name = b.s_name and a.s_sex =
b.s_sex
GROUP BY a.s_name,a.s_sex

-- 31、查询1990年出生的学生名单
select s_name from student where s_birth like '1990%'

-- 32、查询每门课程的平均成绩，结果按平均成绩降序排列，平均成绩相同时，按课程编号升序排列
select c_id,ROUND(AVG(s_score),2) as avg_score from score GROUP BY c_id ORDER
BY avg_score DESC,c_id ASC

-- 33、查询平均成绩大于等于85的所有学生的学号、姓名和平均成绩
select a.s_id,b.s_name,ROUND(avg(a.s_score),2) as avg_score from score a
left join student b on a.s_id=b.s_id GROUP BY s_id HAVING avg_score>=85

-- 34、查询课程名称为"数学"，且分数低于60的学生姓名和分数
select a.s_name,b.s_score from score b join student a on a.s_id=b.s_id where
b.c_id=(
    select c_id from course where c_name = '数学') and b.s_score<60

-- 35、查询所有学生的课程及分数情况：

select a.s_id,a.s_name,
    SUM(case c.c_name when '语文' then b.s_score else 0 end) as '语文',
    SUM(case c.c_name when '数学' then b.s_score else 0 end) as '数学',
    SUM(case c.c_name when '英语' then b.s_score else 0 end) as '英语',
    SUM(b.s_score) as '总分'
from student a left join score b on a.s_id = b.s_id
left join course c on b.c_id = c.c_id
GROUP BY a.s_id,a.s_name

-- 36、查询任何一门课程成绩在70分以上的姓名、课程名称和分数；

```



```

select a.s_name,b.c_name,c.s_score from course b left join score c on
b.c_id = c.c_id
left join student a on a.s_id=c.s_id where c.s_score>=70

-- 37、查询不及格的课程
select a.s_id,a.c_id,b.c_name,a.s_score from score a left join course b on
a.c_id = b.c_id
where a.s_score<60

-- 38、查询课程编号为01且课程成绩在80分以上的学生的学号和姓名：
select a.s_id,b.s_name from score a LEFT JOIN student b on a.s_id = b.s_id
where a.c_id = '01' and a.s_score>80

-- 39、求每门课程的学生人数
select count(*) from score GROUP BY c_id;

-- 40、查询选修"张三"老师所授课程的学生中，成绩最高的学生信息及其成绩

-- 查询老师id
select c_id from course c,teacher d where c.t_id=d.t_id and d.t_name='张三'
-- 查询最高分（可能有相同分数）
select MAX(s_score) from score where c_id='02'
-- 查询信息
select a.*,b.s_score,b.c_id,c.c_name from student a
LEFT JOIN score b on a.s_id = b.s_id
LEFT JOIN course c on b.c_id=c.c_id
where b.c_id =(select c_id from course c,teacher d where c.t_id=d.t_id and
d.t_name='张三')
and b.s_score in (select MAX(s_score) from score where c_id='02')

-- 41、查询不同课程成绩相同的学生的学生编号、课程编号、学生成绩
select DISTINCT b.s_id,b.c_id,b.s_score from score a,score b where a.c_id !=
b.c_id and a.s_score = b.s_score

-- 42、查询每门功成绩最好的前两名
-- 牛逼的写法
select a.s_id,a.c_id,a.s_score from score a
where (select COUNT(1) from score b where b.c_id=a.c_id and
b.s_score>=a.s_score)<=2 ORDER BY a.c_id
-- 43、统计每门课程的学生选修人数（超过5人的课程才统计）。要求输出课程号和选修人数，查询结果按人
数降序排列，若人数相同，按课程号升序排列
select c_id,count(*) as total from score GROUP BY c_id HAVING total>5 ORDER
BY total,c_id ASC

-- 44、检索至少选修两门课程的学生学号
select s_id,count(*) as sel from score GROUP BY s_id HAVING sel>=2

-- 45、查询选修了全部课程的学生信息
select * from student where s_id in(
select s_id from score GROUP BY s_id HAVING count(*)=(select count(*) from
course))

-- 46、查询各学生的年龄
-- 按照出生日期来算，当前月日 < 出生年月的月日则，年龄减一
select s_birth,(DATE_FORMAT(NOW(),'%Y')-DATE_FORMAT(s_birth,'%Y')) -

```

```
(case when DATE_FORMAT(NOW(), '%m%d')>DATE_FORMAT(s_birth, '%m%d') then 0
else 1 end)) as age
from student;

-- 47、查询本周过生日的学生
select * from student where WEEK(DATE_FORMAT(NOW(), '%Y%m%d'))=WEEK(s_birth)
select * from student where
YEARWEEK(s_birth)=YEARWEEK(DATE_FORMAT(NOW(), '%Y%m%d'))

select WEEK(DATE_FORMAT(NOW(), '%Y%m%d'))

-- 48、查询下周过生日的学生
select * from student where WEEK(DATE_FORMAT(NOW(), '%Y%m%d'))+1 =WEEK(s_birth)

-- 49、查询本月过生日的学生
select * from student where MONTH(DATE_FORMAT(NOW(), '%Y%m%d')) =MONTH(s_birth)

-- 50、查询下月过生日的学生
select * from student where MONTH(DATE_FORMAT(NOW(), '%Y%m%d'))+1
=MONTH(s_birth)
```

## 动力节点Java资源库

30+分类 500+G资源 持续更新中



添加小姐姐回复进群  
进学习资源交流群



## 在公众号内回复关键词领取资源

工具

简历

学习路线图

面试题

电子书

技术文档

毕业设计

零基础

公开课

知识卡片

进群

javase

web前端

oracle

jgs

jQuery

javascript

html css

java web

mysql

spring

ssh

ssm

spring boot

springmvc

mybatis

dubbo

redis

git

linux

egov

crm

客服



javazx01



m.wkcto.com

动力节点Java资源库 专注口碑与分享

点个「在看」少个bug