# 1建库

create database tppamltest3

# 2建表

## 1.学生表 Student

学生编号 s\_id主键varchar(20),,非空，默认’’

学生姓名s\_name VARCHAR(20),非空，默认’’

出生年月s\_birth, VARCHAR(20),非空，默认’’

学生性别s\_sex,VARCHAR(10),非空，默认’’

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)

);

*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' , '孙风' , '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');

## 2.课程表 Course

课程编号, c\_id，主键，varchar(20)

课程名称, c\_name VARCHAR(20),非空，默认’’

教师编号，t\_id VARCHAR(20),非空

create table Course(

c\_id varchar(20),

c\_name varchar(20)not null default'',

t\_id varchar(20) not null,

primary key(c\_id)

)

*--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');

## 3.教师表 Teacher

教师编号 t\_id主键 varchar(20)

教师姓名 t\_name varchar(20) 非空 默认’’

create table Teacher(

t\_id varchar(20),

t\_name varchar(20) not null default'',

primary key(t\_id)

);

*-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 Score

学生编号 s\_id联合主键 VARCHAR(20)

课程编号 c\_id联合主键 VARCHAR(20)

分数s\_score int(3)

create table score(

s\_id varchar(20),

c\_id varchar(20),

s\_score int(3),

primary key(s\_id,c\_id)

);

*--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);

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

student.\*, score.s\_score AS 01\_score, c.s\_score AS 02\_score

FROM

student

JOIN

score ON student.s\_id = score.s\_id

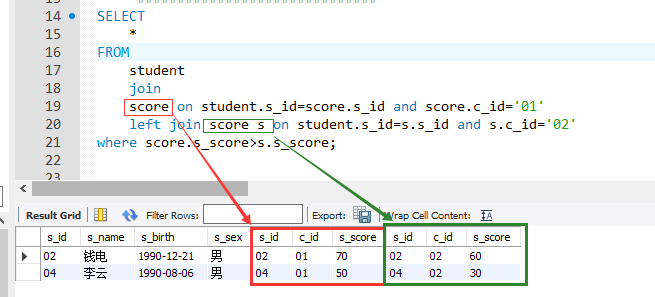
AND score.c\_id = '01'

LEFT JOIN

score c ON student.s\_id = c.s\_id AND c.c\_id = '02'

WHERE

score.s\_score > c.s\_score;



知识点：

Join：

Left join：

1. 查询"01"课程比"02"课程成绩低的学生的信息及课程分数
2. 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（保留两位小数）

SELECT

student.s\_id, student.s\_name, AVG(score.s\_score) AS avg

FROM

student

JOIN

score ON student.s\_id = score.s\_id

GROUP BY student.s\_id , student.s\_name

HAVING AVG(score.s\_score) > 60;

知识点：

Group by：用于结合聚合函数，根据一个或多个列对结果集进行分组

Having：在sql中增加having的原因是，where关键字无法与聚合函数一起使用。Having子句可以让我们筛选分组后的各组数据

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)

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

student.s\_name,student.s\_id,count(score.s\_id) as count\_course,sum(score.s\_score) as sum\_score

from

student

left join score on student.s\_id=score.s\_id

group by student.s\_id;

知识点：

Sum，count

6、查询"李"姓老师的数量

select

count(t\_id)

from

teacher

where

t\_name like '李%'

知识点：

Like的用法

1. 查询学过"张三"老师授课的同学的信息

select

student.\*,score.c\_id

from

student

left join score on student.s\_id=score.s\_id where score.c\_id=

(select course.c\_id from course where course.t\_id=(

select teacher.t\_id from teacher where teacher.t\_name='张三'))；

1. 查询没学过"张三"老师授课的同学的信息

select

student.\*

from

student

where

student.s\_id not in(

select score.s\_id from score where score.c\_id=(

select course.c\_id from course where course.t\_id=(

select teacher.t\_id from teacher where teacher.t\_name='张三')));

1. 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';

select

student.\*,score.s\_score as 02\_course,a.s\_score as 01\_course

from

student

left join score on student.s\_id=score.s\_id and score.c\_id='02'

left join score a on student.s\_id=a.s\_id and a.c\_id='01'

where score.s\_score is not null and a.s\_score is not null;

1. 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

select

student.\*,score.s\_score as 01score,a.s\_score as 02score

FROM

student

left join score on student.s\_id=score.s\_id and score.c\_id='01'

left join score a on student.s\_id=a.s\_id and a.c\_id='02'

where score.s\_score is not null and a.s\_score is null;

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')

1. 查询没有学全所有课程的同学的信息

select

student.\*,score.s\_score as 01score,b.s\_score as 02score,c.s\_score as 03score

from

student

left join score on student.s\_id=score.s\_id and score.c\_id='01'

left join score b on student.s\_id=b.s\_id and b.c\_id='02'

left join score c on student.s\_id=c.s\_id and c.c\_id='03'

where score.s\_score is not null and b.s\_score is not null and c.s\_score is not null

1. 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

student.\*

from

student

left join score on score.s\_id=student.s\_id and score.c\_id in (

select score.c\_id from score where s\_id='01')

where score.s\_score is not null

group by student.s\_id;

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')

);

知识点：

Distinct：对查询出来的字段，去重

1. 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 （不会）

select a.\* from student a where a.s\_id in(

select distinct s\_id from score where s\_id!='01' and c\_id in(select c\_id from score where s\_id='01')

group by s\_id

having count(1)=(select count(1) from score where s\_id='01'));

1. 检索"01"课程分数小于60，按分数降序排列的学生信息

select

student.\*,score.c\_id ,score.s\_score

from

student

left join score on student.s\_id=score.s\_id and score.c\_id='01' and score.s\_score<60

order by score.s\_score desc

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;

1. 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 b.s\_score from score b where a.s\_id=b.s\_id and b.c\_id='01') as 语文,

(select c.s\_score from score c where a.s\_id=c.s\_id and c.c\_id='02') as 数学,

(select d.s\_score from score d where a.s\_id=d.s\_id and d.c\_id='02') as 英语,

round(avg(a.s\_score),2) as ave\_score

from

score a

group by a.s\_id

order by ave\_score desc;

# 聚合函数

•1、AVG：求平均值

•2、COUNT

•3、MIN / MAX

•4 SUM