[**Mysql进行复杂查询**](http://www.cnblogs.com/pyramid1001/p/5990194.html)

1.查询“生物”课程比“物理”课程成绩高的所有学生的学号；

 思路： （1）获取所有选了 生物 课程的学生的成绩（学号，成绩） --临时表

　　    （2）获取所有选了 物理 课程的学生的成绩（学号，成绩） --临时表

　　　（3）根据学号连接两张临时表（学号，生物成绩，物理成绩），加条件进行查询

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

[复制代码](javascript:void(0);)

SELECT

A.student\_id AS 学号,

sw AS 生物,

wl AS 物理

FROM

(

SELECT

student\_id,

num AS sw

FROM

score

LEFT JOIN course ON score.course\_id = course.cid

WHERE

course.cname = '生物'

) AS A

LEFT JOIN (

SELECT

student\_id,

num AS wl

FROM

score

LEFT JOIN course ON score.course\_id = course.cid

WHERE

course.cname = '物理'

) AS B ON A.student\_id = B.student\_id

WHERE

sw >

IF (isnull(wl), **0**, wl);

[复制代码](javascript:void(0);)

2.查询平均成绩大于60分的同学的学号和平均成绩；

思路：（1）根据学号分组

　　  （2）使用avg（）聚合函数计算平均成绩

　　  （3）通过having对平均成绩进行筛选

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

SELECT student\_id,avg(num) FROM score

LEFT JOIN course

ON score.student\_id=course.cid

GROUP BY student\_id

HAVING avg(num)>**60**

3.查询所有同学的学号、姓名、选课数、总成绩；

思路：根据学号分组，使用count（）对选课数计数，sum（）计算总成绩

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

[复制代码](javascript:void(0);)

SELECT

student\_id,

sname,

count(student\_id),

sum(num)

FROM

score LEFT JOIN student ON score.student\_id = student.sid

GROUP BY student\_id

[复制代码](javascript:void(0);)

4.查询姓“李”的老师的个数；

思路：使用like及通配符匹配，count（）进行计数

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

SELECT count(tid) 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；

思路：（1）连接成绩表 课程表 教师表得到选了李平老师课程的学生

         （2）再通过学生表筛选结果

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

[复制代码](javascript:void(0);)

SELECT sid,sname FROM student

WHERE sid not IN

(SELECT student\_id FROM

(SELECT cid,teacher\_id,student\_id,course\_id

FROM score

LEFT JOIN course ON score.course\_id=course.cid

) AS A

LEFT JOIN teacher

ON A.teacher\_id=teacher.tid where teacher.tname='李平老师' GROUP BY student\_id

)

[复制代码](javascript:void(0);)

6.查询学过“001”并且也学过编号“002”课程的同学的学号、姓名；

思路：（1）筛选出学过001课程的学生或学过002课程的学生

　　  （2）根据学生分组，如果学生数量等于2，则该学生选择了以上两门课程

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

SELECT student.sid,student.sname FROM

(SELECT student\_id,count(student\_id) FROM score LEFT JOIN course ON score.course\_id=course.cid WHERE course.cid='001' or course.cid='002' GROUP BY student\_id HAVING count(student\_id)>**1** ) AS A

LEFT JOIN student ON A.student\_id=student.sid;

7.查询学过“李平”老师所教的所有课的同学的学号、姓名；

思路：（1）查询李平老师所教的课程

         （2）在成绩表中筛选出学生选择的课程 in 李平老师的课程

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

[复制代码](javascript:void(0);)

SELECT student\_id,sname FROM

(SELECT student\_id FROM score

WHERE course\_id IN

(SELECT cid FROM teacher LEFT JOIN course ON teacher.tid=course.teacher\_id WHERE tname='李平老师')

GROUP BY student\_id

) AS B

LEFT JOIN student

ON B.student\_id=student.sid

[复制代码](javascript:void(0);)

8.查询课程编号“002”的成绩比课程编号“001”课程低的所有同学的学号、姓名；

思路：（1）分别获取选择了 课程001和002的学生和成绩；

　　 （2）连接两张表，筛选出001的成绩大于002成绩的学生

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

[复制代码](javascript:void(0);)

SELECT student\_id,sname FROM

(SELECT A.student\_id FROM

(SELECT student\_id,num FROM score WHERE course\_id=**001**) AS A

LEFT JOIN

(SELECT student\_id,num FROM score WHERE course\_id=**002**) AS B

ON A.student\_id=B.student\_id

WHERE A.num>B.num) AS C

LEFT JOIN student

ON C.student\_id=student.sid

[复制代码](javascript:void(0);)

9.查询有课程成绩小于60分的同学的学号、姓名；

思路：（1）筛选出成绩小于60的学生，并通过学生分组   --临时表

　　  （2）在学生表中筛选出 in 临时表中的学生

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

SELECT sid,sname FROM student WHERE sid IN

(SELECT student\_id FROM score WHERE num<**60** GROUP BY student\_id)

10.查询没有学全所有课的同学的学号、姓名；

思路：（1）统计出总课程数

         （2）成绩表中，通过学生分组，统计出每个学生的课程数，如果课程数等于总课程数，则表示选择了所有课程

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

SELECT sid,sname FROM student WHERE sid not IN

(SELECT student\_id

FROM score

GROUP BY student\_id HAVING count(course\_id) = (SELECT count(cid) FROM course))

11.查询至少有一门课与学号为“001”的同学所学课程相同的同学的学号和姓名；

思路：（1）查找学号001同学所学的所有课程　　--临时表

　　  （2）其他学生所学的课程如果在临时表中，则符合条件

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

SELECT student\_id,sname FROM student LEFT JOIN score ON score.student\_id=student.sid

WHERE course\_id in (SELECT course\_id FROM score WHERE student\_id=**001**) AND student\_id != **001**

GROUP BY student\_id

12.查询至少学过学号为“001”同学所选课程中任意一门课的**其他同学**学号和姓名；

13.查询和“002”号的同学学习的课程完全相同的其他同学学号和姓名；

14.删除学习“叶平”老师课的成绩表记录；

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

DELETE FROM score WHERE course\_id IN

(SELECT cid FROM course LEFT JOIN teacher ON course.teacher\_id=teacher.tid WHERE tname='叶平老师')

15.向成绩表中插入一些记录，这些记录要求符合以下条件：①没有上过编号“002”课程的同学学号；②插入“002”号课程的平均成绩；

16.按平均成绩从低到高显示所有学生的“语文”、“数学”、“英语”三门的课程成绩，按如下形式显示： 学生ID,语文,数学,英语,有效课程数,有效平均分；

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

[复制代码](javascript:void(0);)

select sc.student\_id,

(select num from score left join course on score.course\_id = course.cid where course.cname = "生物" and score.student\_id=sc.student\_id) as sy,

(select num from score left join course on score.course\_id = course.cid where course.cname = "物理" and score.student\_id=sc.student\_id) as wl,

(select num from score left join course on score.course\_id = course.cid where course.cname = "体育" and score.student\_id=sc.student\_id) as ty,

count(sc.course\_id),

avg(sc.num)

from score as sc

group by student\_id desc

[复制代码](javascript:void(0);)

17.查询各科成绩最高和最低的分：以如下形式显示：课程ID，最高分，最低分；

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

SELECT course\_id,max(num) as 最高分,min(num) as 最低分

FROM score

GROUP BY course\_id

18.按各科平均成绩从低到高和及格率的百分数从高到低顺序；

思路：三元运算（三目运算），case .. when .. then .. else .. end

SELECT course\_id,avg(num) AS 平均分,sum(CASE WHEN score.num>**60** THEN **1** ELSE **0** END)/count(**1**)\***100** AS 及格率 FROM score GROUP BY course\_id

ORDER BY 平均分 ASC,及格率 DESC

19.课程平均分从高到低显示（显示任课老师）；

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

SELECT course\_id,avg(num),teacher.tname FROM score LEFT JOIN course ON score.course\_id=course.cid LEFT JOIN teacher ON course.teacher\_id=teacher.tid

GROUP BY course\_id

ORDER BY avg(num) DESC

20.查询各科成绩前三名的记录:(不考虑成绩并列情况) ;

21.查询每门课程被选修的学生数；

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

SELECT course\_id,count(student\_id) FROM score

GROUP BY course\_id

22.查询出只选修了一门课程的全部学生的学号和姓名；

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

SELECT student\_id,student.sname FROM score LEFT JOIN student ON score.student\_id=student.sid

GROUP BY student\_id

HAVING count(student\_id)=**1**

23.查询男生、女生的人数；

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

SELECT

(SELECT count(**1**) FROM student WHERE gender='男') AS 男,

(SELECT count(**1**) FROM student WHERE gender='女') As 女

24.查询姓“张”的学生名单；

SELECT \* FROM student WHERE student.sname LIKE '张%'

25.查询同名同姓学生名单，并统计同名人数；

SELECT sname,count(sname) FROM student GROUP BY sname HAVING count(sname)>**1**

26.查询每门课程的平均成绩，结果按平均成绩升序排列，平均成绩相同时，按课程号降序排列；

SELECT course\_id,avg(num) FROM score GROUP BY course\_id ORDER BY course\_id ASC,course\_id DESC

27.查询平均成绩大于85的所有学生的学号、姓名和平均成绩；

SELECT student\_id,sname,avg(num) FROM score LEFT JOIN student ON score.student\_id=student.sid

GROUP BY student\_id HAVING avg(num)>**85**

28.查询课程名称为“数学”，且分数低于60的学生姓名和分数；

SELECT sname,num FROM score LEFT JOIN course ON score.course\_id=course.cid LEFT JOIN student ON score.student\_id=student.sid WHERE cname='数学' AND num>**60**

29.查询课程编号为003且课程成绩在80分以上的学生的学号和姓名；

SELECT student\_id,sname FROM score LEFT JOIN student ON score.student\_id=student.sid WHERE course\_id=**003** AND num>**80**

30.求选了课程的学生人数

select count(distinct student\_id) from score

31.查询选修“杨艳”老师所授课程的学生中，成绩最高的学生姓名及其成绩；

思路：根据学生排序，成绩按从大到小排序，limit取最高的成绩

[复制代码](javascript:void(0);)

SELECT sname,max(num) FROM score LEFT JOIN course ON score.course\_id=course.cid LEFT JOIN teacher ON course.teacher\_id=teacher.tid

LEFT JOIN student ON score.student\_id=student.sid

WHERE tname='张磊老师'

GROUP BY student\_id

ORDER BY max(num) DESC

LIMIT **1**

[复制代码](javascript:void(0);)

32.查询各个课程及相应的选修人数；

SELECT cid,cname,count(student\_id) FROM score LEFT JOIN course ON score.course\_id=course.cid

GROUP BY course\_id

33.查询不同课程但成绩相同的学生的学号、课程号、学生成绩；

select DISTINCT s1.course\_id,s2.course\_id,s1.num,s2.num from

score as s1, score as s2 where s1.num = s2.num and s1.course\_id != s2.course\_id;

34.查询每门课程成绩最好的前两名；

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

[复制代码](javascript:void(0);)

SELECT \* FROM score LEFT JOIN

(SELECT course\_id,

(SELECT num FROM score WHERE score.course\_id=A.course\_id ORDER BY num DESC LIMIT **0**,**1**) AS '第一名',

(SELECT num FROM score WHERE score.course\_id=A.course\_id ORDER BY num DESC LIMIT **1**,**1**) AS '第二名'

FROM score AS A

GROUP BY course\_id) AS B

ON score.course\_id=B.course\_id

[复制代码](javascript:void(0);)

35.检索至少选修两门课程的学生学号；

思路：根据学号分组，统计

SELECT student\_id,count(course\_id) FROM score GROUP BY student\_id HAVING count(course\_id)>=**2**

36.查询全部学生都选修的课程的课程号和课程名；

思路：从学生表中统计出学生总数，在成绩表中根据课程分组，如果选择没门课程的人数等于学生总数，则符合

SELECT course\_id,cname FROM score LEFT JOIN course ON score.course\_id=course.cid GROUP BY course\_id HAVING count(student\_id)=

(SELECT count(sid) FROM student)

37.查询没学过“叶平”老师讲授的任一门课程的学生姓名；

SELECT sname FROM score LEFT JOIN student ON score.student\_id=student.sid WHERE course\_id NOT IN

(SELECT cid FROM teacher LEFT JOIN course ON course.teacher\_id=teacher.tid WHERE tname='叶平老师')

38.查询两门以上不及格课程的同学的学号及其平均成绩；

思路：通过学生分组，筛选出不及格课程数

SELECT student\_id,avg(num) FROM score WHERE num<**60** GROUP BY student\_id HAVING count(**1**)>**2**

39.检索“004”课程分数小于60，按分数降序排列的同学学号；

SELECT student\_id FROM score WHERE course\_id=**004** AND num<**60** ORDER BY num DESC

40.删除“002”同学的“001”课程的成绩；

DELETE FROM score WHERE student\_id=**002** AND course\_id=**001**