Mysql练习题

**Class表的定义**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **字段名** | **字段描述** | **数据类型** | **主键** | **外键** | **非空** | **唯一** | **自增** |
| class\_id | 编号 | INT(10) | 是 | 否 | 是 | 是 | 是 |
| class\_name | 班级名称 | VARCHAR(64) | 否 | 否 | 是 | 否 | 否 |

INSERT INTO `class` VALUES ('1', '三年二班'), ('2', '三年三班'), ('3', '一年二班'), ('4', '二年九班');

**Subject表的定义**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **字段名** | **字段描述** | **数据类型** | **主键** | **外键** | **非空** | **唯一** | **自增** |
| subject\_id | 编号 | INT(10) | 是 | 否 | 是 | 是 | 是 |
| subject\_name | 班级名称 | VARCHAR(64) | 否 | 否 | 是 | 否 | 否 |
| teacher\_id | 教师id | INT(10) | 否 | 否 | 否 | 否 | 否 |

INSERT INTO `course` VALUES ('1', '生物', '1'), ('2', '物理', '2'), ('3', '体育', '3'), ('4', '美术', '2');

**Score表的定义**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **字段名** | **字段描述** | **数据类型** | **主键** | **外键** | **非空** | **唯一** | **自增** |
| score\_id | 编号 | INT(10) | 是 | 否 | 是 | 是 | 是 |
| subject\_id | 课程id | INT(10) | 否 | 否 | 是 | 否 | 否 |
| student\_id | 学生id | INT(10) | 否 | 否 | 否 | 否 | 否 |
| score | 分数 | INT(10) | 否 | 否 | 否 | 否 | 否 |

INSERT INTO `score` VALUES ('1', '1', '1', '10'), ('2', '1', '2', '9'), ('5', '1', '4', '66'), ('6', '2', '1', '8'), ('8', '2', '3', '68'), ('9', '2', '4', '99'), ('10', '3', '1', '77'), ('11', '3', '2', '66'), ('12', '3', '3', '87'), ('13', '3', '4', '99'), ('14', '4', '1', '79'), ('15', '4', '2', '11'), ('16', '4', '3', '67'), ('17', '4', '4', '100'), ('18', '5', '1', '79'), ('19', '5', '2', '11'), ('20', '5', '3', '67'), ('21', '5', '4', '100'), ('22', '6', '1', '9'), ('23', '6', '2', '100'), ('24', '6', '3', '67'), ('25', '6', '4', '100'), ('26', '7', '1', '9'), ('27', '7', '2', '100'), ('28', '7', '3', '67'), ('29', '7', '4', '88'), ('30', '8', '1', '9'), ('31', '8', '2', '100'), ('32', '8', '3', '67'), ('33', '8', '4', '88'), ('34', '9', '1', '91'), ('35', '9', '2', '88'), ('36', '9', '3', '67'), ('37', '9', '4', '22'), ('38', '10', '1', '90'), ('39', '10', '2', '77'), ('40', '10', '3', '43'), ('41', '10', '4', '87'), ('42', '11', '1', '90'), ('43', '11', '2', '77'), ('44', '11', '3', '43'), ('45', '11', '4', '87'), ('46', '12', '1', '90'), ('47', '12', '2', '77'), ('48', '12', '3', '43'), ('49', '12', '4', '87'), ('52', '13', '3', '87');

**Student表的定义**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **字段名** | **字段描述** | **数据类型** | **主键** | **外键** | **非空** | **唯一** | **自增** |
| student\_id | 编号 | INT(10) | 是 | 否 | 是 | 是 | 是 |
| sex | 性别 | VARCHAR(64) | 否 | 否 | 是 | 否 | 否 |
| class\_id | 班级id | INT(10) | 否 | 否 | 否 | 否 | 否 |
| student\_name | 学生姓名 | VARCHAR(64) | 否 | 否 | 否 | 否 | 否 |

INSERT INTO `student` VALUES ('1', '男', '1', '理解'), ('2', '女', '1', '钢蛋'), ('3', '男', '1', '张三'), ('4', '男', '1', '张一'), ('5', '女', '1', '张二'), ('6', '男', '1', '张四'), ('7', '女', '2', '铁锤'), ('8', '男', '2', '李三'), ('9', '男', '2', '李一'), ('10', '女', '2', '李二'), ('11', '男', '2', '李四'), ('12', '女', '3', '如花'), ('13', '男', '3', '刘三'), ('14', '男', '3', '刘一'), ('15', '女', '3', '刘二'), ('16', '男', '3', '刘四');

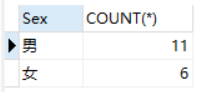
**Teacher表的定义**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **字段名** | **字段描述** | **数据类型** | **主键** | **外键** | **非空** | **唯一** | **自增** |
| teacher\_id | 编号 | INT(10) | 是 | 否 | 是 | 是 | 是 |
| teacher\_name | 姓名 | VARCHAR(64) | 否 | 否 | 是 | 否 | 否 |

INSERT INTO `teacher` VALUES ('1', '张磊老师'), ('2', '李平老师'), ('3', '刘海燕老师'), ('4', '朱云海老师'), ('5', '李杰老师');

1. 查询男生、女生的人数；

SELECT Sex,COUNT(\*) FROM student GROUP BY Sex



1. 查询姓“张”的学生名单；

SELECT student\_name FROM student WHERE student\_name LIKE '张%'



1. 课程平均分从高到低显示

SELECT

subject\_id,AVG(score)

FROM

score

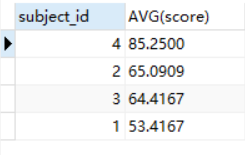
GROUP BY

subject\_id

ORDER BY

AVG(score)

DESC;



1. 查询有课程成绩小于60分的同学的学号、姓名；

SELECT

DISTINCT st.student\_id,st.student\_name

FROM

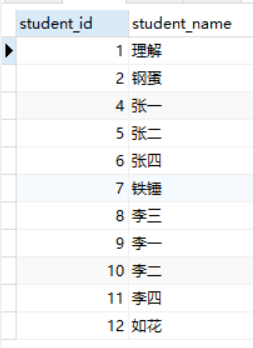
student st,score sc

WHERE

st.student\_id=sc.student\_id

AND

score<60



1. 查询至少有一门课与学号为1的同学所学课程相同的同学的学号和姓名；

SELECT

DISTINCT st.student\_id,st.student\_name

FROM

student st,score sc

WHERE

st.student\_id=sc.student\_id

AND

sc.subject\_id

IN

(SELECT

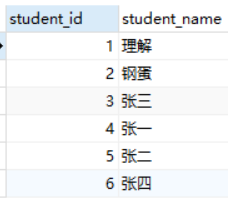
subject\_id

FROM

score

WHERE

student\_id=1)



1. 查询出只选修了一门课程的全部学生的学号和姓名；

SELECT

student\_id,student\_name

FROM

student

WHERE

student\_id

IN

(SELECT

student\_id

FROM

(SELECT

student\_id,COUNT(\*) c

FROM

score

GROUP BY

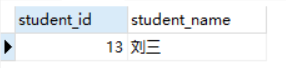
student\_id

) a

WHERE

c=1

)



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

SELECT

subject\_id '课程ID',MAX(score) '最高分',MIN(score) '最低分'

FROM

score

GROUP BY

subject\_id



1. 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；

SELECT

DISTINCT sc1.student\_id,st.student\_name

FROM

score sc1,student st

WHERE

subject\_id=1

AND

score>

(SELECT

score

FROM

score sc2

WHERE

subject\_id=2

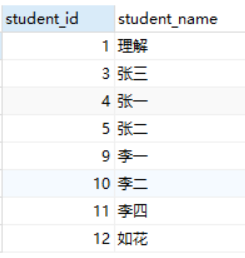
AND

sc2.student\_id=sc1.student\_id

)

AND

sc1.student\_id=st.student\_id



1. 查询“生物”课程比“物理”课程成绩高的所有学生的学号；

SELECT DISTINCT

sc1.student\_id

FROM

score sc1,subject sj

WHERE

subject\_name='生物'

AND

score>

(SELECT

score

FROM

score sc2,subject sj2

WHERE

subject\_name='物理'

AND

sc2.student\_id=sc1.student\_id

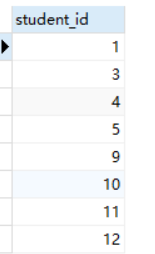
AND

sc2.subject\_id=sj2.subject\_id

)

AND

sc1.subject\_id=sj.subject\_id



1. 查询平均成绩大于60分的同学的学号和平均成绩;

SELECT

student\_id

FROM

(SELECT

student\_id,AVG(score) avg

FROM

score

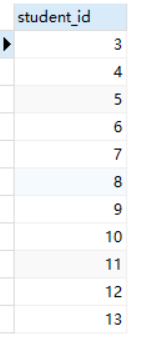
GROUP BY

student\_id

) a

WHERE

avg>60



11.查询所有同学的学号、姓名、选课数、总成绩；

SELECT

stu.student\_id,

student\_name,

sum\_num,

count\_stu

FROM

student stu

LEFT JOIN (

SELECT

sum(score) sum\_num,

count(subject\_id) count\_stu,

student\_id

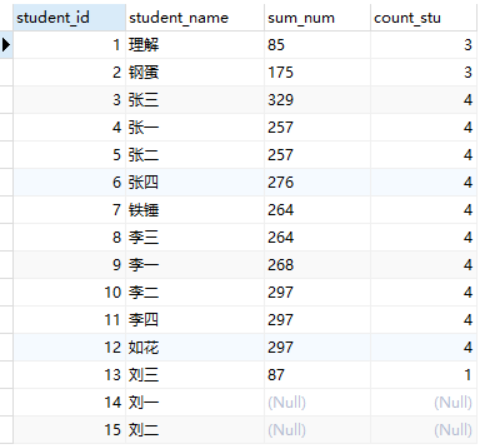
FROM

score

GROUP BY

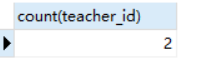
student\_id

) t2 ON stu.student\_id = t2.student\_id



1. 查询姓“李”的老师的个数；

select count(teacher\_id) from teacher where teacher\_name like '李%'



13.查询没学过“张磊老师”课的同学的学号、姓名；

SELECT

student\_id,student\_name

FROM

student

WHERE

student\_id

NOT IN

(

SELECT

sc.student\_id

FROM

teacher tc,subject sj,score sc

WHERE

tc.teacher\_id=sj.teacher\_id

AND

sj.subject\_id=sc.subject\_id

AND

tc.teacher\_name='张磊老师'

)

14.查询学过“1”并且也学过编号“2”课程的同学的学号、姓名；

SELECT

sc.student\_id,st.student\_name

FROM

score sc,student st

WHERE

sc.subject\_id=1

AND

sc.student\_id=st.student\_id

AND

sc.student\_id

IN

(SELECT

student\_id

FROM

score

WHERE

subject\_id=2)

15.查询学过“李平老师”所教的所有课的同学的学号、姓名；

SELECT

student\_id,student\_name

FROM

student

WHERE

student\_id

IN

(

SELECT

sc.student\_id

FROM

teacher tc,subject sj,score sc

WHERE

tc.teacher\_id=sj.teacher\_id

AND

sj.subject\_id=sc.subject\_id

AND

tc.teacher\_name='李平老师'

)

16.查询没有学全所有课的同学的学号、姓名；

SELECT

student\_id,student\_name

FROM

student

WHERE

student\_id

IN

(SELECT

student\_id

FROM

(SELECT

student\_id,COUNT(\*) c

FROM

score

GROUP BY

student\_id) a

WHERE

c!=4

)



17.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；

select student\_id from score where student\_id not in (

select student\_id from score where subject\_id not in (select subject\_id from score where student\_id=2)

) and student\_id !=2

group by student\_id

having count(subject\_id)= (select count(subject\_id) from score where student\_id=2);

18.删除学习“叶平”老师课的SC表记录；

19.向SC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；

20.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；

SELECT

stu.student\_id 学生ID,

t2.score 物理,

t1.score 生物,

t3.score 体育,

t4.score 美术,

t5.count\_subject 有效课程数,

t5.avg\_score 有效平均分

FROM

student stu

LEFT JOIN (

SELECT

student\_id,

score

FROM

score

WHERE

subject\_id = (

SELECT

subject\_id

FROM

`subject`

WHERE

subject\_name = '生物'

)

) t1 ON stu.student\_id = t1.student\_id

LEFT JOIN (

SELECT

student\_id,

score

FROM

score

WHERE

subject\_id = (

SELECT

subject\_id

FROM

`subject`

WHERE

subject\_name = '物理'

)

) t2 ON stu.student\_id = t2.student\_id

LEFT JOIN (

SELECT

student\_id,

score

FROM

score

WHERE

subject\_id = (

SELECT

subject\_id

FROM

`subject`

WHERE

subject\_name = '体育'

)

) t3 ON stu.student\_id = t3.student\_id

LEFT JOIN (

SELECT

student\_id,

score

FROM

score

WHERE

subject\_id = (

SELECT

subject\_id

FROM

`subject`

WHERE

subject\_name = '美术'

)

) t4 ON stu.student\_id = t4.student\_id

LEFT JOIN (

SELECT

student\_id,

avg(score) avg\_score,

count(score) count\_subject

FROM

score

GROUP BY

student\_id

) t5 ON stu.student\_id = t5.student\_id



21.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；

SELECT

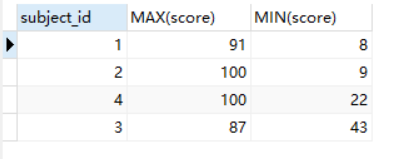
subject\_id,MAX(score),MIN(score)

FROM

score

GROUP BY

subject\_id



22.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；

23.查询各科成绩前三名的记录:(不考虑成绩并列情况)

SELECT

subject\_id,score

FROM

score sc1

WHERE

(SELECT

COUNT(\*)

FROM

score sc2

WHERE

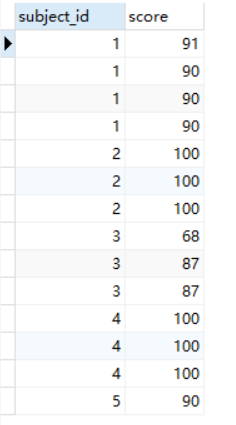
sc1.score<sc2.score

AND

sc1.subject\_id=sc2.subject\_id

)<3

ORDER BY subject\_id



24.查询每门课程被选修的学生数；

SELECT

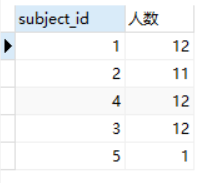
subject\_id,COUNT(\*) 人数

FROM

score

GROUP BY

subject\_id



25.查询同名同姓学生名单，并统计同名人数；

SELECT

student\_name,c

FROM

(SELECT

student\_name,COUNT(\*) c

FROM

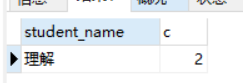
student

GROUP BY

student\_name) a

WHERE

c>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

subject\_id,AVG(score)

FROM

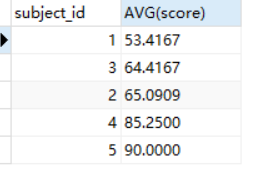
score

GROUP BY

subject\_id

ORDER BY

AVG(score) ASC,subject\_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

st.student\_id,student\_name,avg

FROM

(SELECT

student\_id,AVG(score) avg

FROM

score

GROUP BY

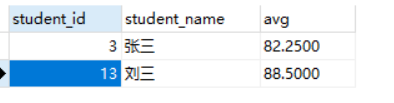
student\_id

HAVING

AVG(score)>80) a,student st

WHERE

a.student\_id=st.student\_id



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

student\_name,score

FROM

score sc,student st

WHERE

subject\_id

in

(SELECT

subject\_id

FROM

subject

WHERE

subject\_name='数学'

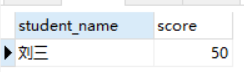
)

AND

score<60

AND

sc.student\_id=st.student\_id



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,student\_name

FROM

student

WHERE

student\_id

IN

(SELECT

student\_id

FROM

score

WHERE

score>80

AND

subject\_id=003)



30.求选了课程的学生人数

SELECT

COUNT(\*)

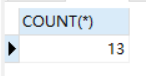
FROM

(SELECT

DISTINCT student\_id

FROM

score) a



31.查询选修“李平”老师所授课程的学生中，成绩最高的学生姓名及其成绩；

SELECT

student.student\_name,

t1.score

FROM

(

SELECT DISTINCT

student\_id,

score

FROM

score

WHERE

score = (

SELECT

max(score)

FROM

score

WHERE

subject\_id IN (

SELECT

subject\_id

FROM

`subject`

WHERE

teacher\_id IN (

SELECT

teacher\_id

FROM

teacher

WHERE

teacher\_name = '李平老师'

)

)

)

AND subject\_id IN (

SELECT

subject\_id

FROM

`subject`

WHERE

teacher\_id IN (

SELECT

teacher\_id

FROM

teacher

WHERE

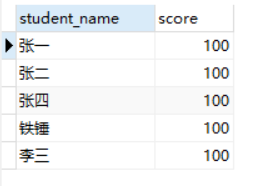
teacher\_name = '李平老师'

)

)

) t1

LEFT JOIN student ON t1.student\_id = student.student\_id



32.查询各个课程及相应的选修人数；

SELECT

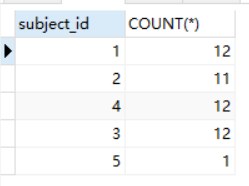
subject\_id,COUNT(\*)

FROM

score

GROUP BY

subject\_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

student\_id,subject\_id,score

FROM

score sc1

WHERE

(

SELECT

COUNT(\*)

FROM

score sc2

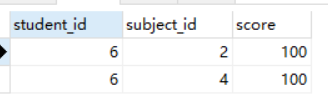
WHERE

sc1.score=sc2.score

AND

sc1.student\_id=sc2.student\_id

)>1



34.查询每门课程成绩最好的前两名；

SELECT

subject\_id,score

FROM

score sc1

WHERE

(SELECT

COUNT(\*)

FROM

score sc2

WHERE

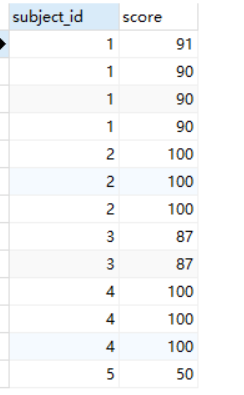
sc1.score<sc2.score

AND

sc1.subject\_id=sc2.subject\_id

)<2

ORDER BY subject\_id



35.检索至少选修两门课程的学生学号；

SELECT

student\_id

FROM

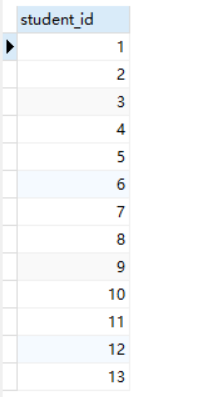
score

GROUP BY

student\_id

HAVING

COUNT(\*)>=2



36.查询全部学生都选修的课程的课程号和课程名；

SELECT

sj.subject\_id,subject\_name

FROM

score sc

LEFT JOIN

subject sj

ON

sc.subject\_id=sj.subject\_id

AND

sc.subject\_id

IN

(SELECT

subject\_id

FROM

score

GROUP BY

subject\_id

HAVING

COUNT(\*)=(SELECT COUNT(\*) 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

student\_name

FROM

student

WHERE

student\_id NOT IN (

SELECT DISTINCT

student\_id

FROM

score

WHERE

subject\_id IN (

SELECT

subject\_id

FROM

`subject`

WHERE

teacher\_id IN (

SELECT

teacher\_id

FROM

teacher

WHERE

teacher\_name = '李平老师'

)

)

);



38.查询两门以上不及格课程的同学的学号及其平均成绩；

SELECT

student\_id,AVG(score)

FROM

score sc1

WHERE

score<60

GROUP BY

student\_id

HAVING

COUNT(\*)>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

subject\_id=4

AND

score<60

ORDER BY

score

DESC



40.删除“002”同学的“001”课程的成绩；

DELETE FROM

score

WHERE

student\_id=2

OR

student\_id=1