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；
2. 编写代码：

SELECT

sex '性别',

COUNT(\*) '人数'

FROM student

GROUP BY sex

（2）运行结果如下图：



1. 查询姓“张”的学生名单；
2. 编写代码：

SELECT

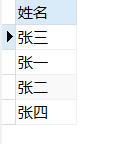
student\_name '姓名'

FROM student

WHERE

student\_name LIKE '张%'

1. 运行结果如下图：



1. 课程平均分从高到低显示
2. 编写代码：

SELECT

subject\_id '课程编号',

avg(score) '平均分'

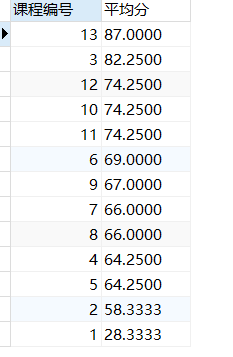
FROM

score

GROUP BY subject\_id

ORDER BY 平均分 DESC

1. 运行结果如下图：



1. 查询有课程成绩小于60分的同学的学号、姓名；
2. 编写代码：

SELECT DISTINCT

stu.student\_id '学号',

stu.student\_name '姓名'

FROM

student stu LEFT JOIN score sc ON stu.student\_id = sc.student\_id

WHERE

sc.score < 60

1. 运行结果如下图：



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；
2. 编写代码：

SELECT DISTINCT

stu.student\_id '学号',

stu.student\_name '姓名'

FROM

student stu,score sc

WHERE

stu.student\_id = sc.student\_id

AND

sc.subject\_id in(SELECT subject\_id '课程编号'

FROM score

WHERE student\_id = 1 )

1. 运行结果如下图：



1. 查询出只选修了一门课程的全部学生的学号和姓名；
2. 编写代码：

SELECT

stu.student\_id '学号',

stu.student\_name '姓名'

FROM

student stu,

(SELECT

COUNT(subject\_id) '课程数',

student\_id

FROM score

GROUP BY student\_id

) cnt

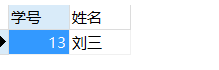
WHERE

stu.student\_id = cnt.student\_id

AND

cnt.课程数 = 1

1. 运行结果如下图：



1. 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；
2. 编写代码：

SELECT

sc\_max.subject\_id '课程ID',

max '最高分',

min '最低分'

FROM

(SELECT subject\_id,max(score) as max

FROM score

GROUP BY subject\_id

) sc\_max,

(SELECT subject\_id,min(score) as min

FROM score

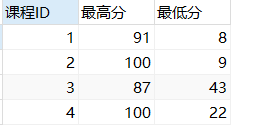
GROUP BY subject\_id

) sc\_min

WHERE

sc\_max.subject\_id = sc\_min.subject\_id

1. 运行结果如下图：



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；
2. 编写代码：

SELECT

student\_id '学号',

student\_name '姓名'

FROM

student

WHERE

student\_id IN

(SELECT t1.student\_id

FROM

(SELECT student\_id,score

FROM score

WHERE subject\_id = 2) t2

INNER JOIN

(SELECT student\_id,score

FROM score

WHERE subject\_id = 1) t1

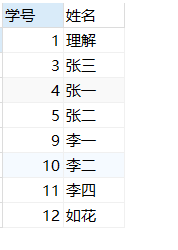
ON

t1.student\_id = t2.student\_id

WHERE

t2.score < t1.score);

1. 运行结果如下图：



1. 查询“生物”课程比“物理”课程成绩高的所有学生的学号；
2. 编写代码：

SELECT

student\_id '学号',

student\_name '姓名'

FROM

student

WHERE

student\_id IN

(SELECT t1.student\_id

FROM

(SELECT student\_id,score

FROM score sc INNER JOIN `subject` sub

ON sc.subject\_id = sub.subject\_id

WHERE subject\_name = '物理') t2

INNER JOIN

(SELECT student\_id,score

FROM score sc INNER JOIN `subject` sub

ON sc.subject\_id = sub.subject\_id

WHERE subject\_name = '生物') t1

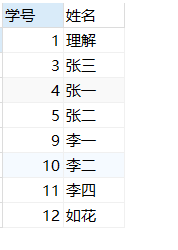
ON

t1.student\_id = t2.student\_id

WHERE

t2.score < t1.score);

1. 运行结果如下图：



1. 查询平均成绩大于60分的同学的学号和平均成绩;
2. 编写代码：

SELECT

student\_id '学号',

avg(score) '平均成绩'

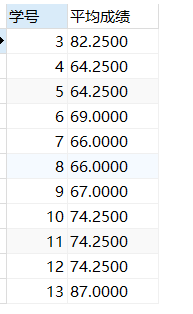
FROM

score

GROUP BY student\_id

HAVING avg(score) > 60

1. 运行结果如下图：



1. 查询所有同学的学号、姓名、选课数、总成绩；
2. 编写代码：

SELECT

stu.student\_id '学号',

student\_name '姓名',

COUNT(\*) '选课数',

SUM(score) '总成绩'

FROM

student stu INNER JOIN score sc

ON stu.student\_id = sc.student\_id

GROUP BY stu.student\_id

1. 运行结果如下图



1. 查询姓“李”的老师的个数；

（1）编写代码：

SELECT

COUNT(\*) '姓李老师个数'

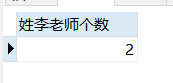
FROM

teacher

where

teacher\_name LIKE '李%'

（2）运行结果如下图：



1. 查询没学过“张磊老师”课的同学的学号、姓名；
2. 编写代码：

SELECT

student\_id '学号',

student\_name '姓名'

FROM

student

WHERE

student\_id NOT IN (

SELECT student\_id

FROM score sc INNER JOIN

(SELECT subject\_id,teacher\_name

FROM `subject` sub,teacher tea

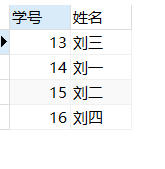
WHERE sub.teacher\_id = tea.teacher\_id) t1

ON sc.subject\_id = t1.subject\_id

WHERE

t1.teacher\_name = '张磊老师')

1. 运行结果如下图：



1. 查询学过“1”并且也学过编号“2”课程的同学的学号、姓名；
2. 编写代码：

SELECT

student\_id '学号',

student\_name '姓名'

FROM

student

WHERE

student\_id IN(

SELECT t1.student\_id

FROM

(SELECT student\_id

FROM score

WHERE subject\_id = 1) t1

INNER JOIN

(SELECT student\_id

FROM score

WHERE subject\_id = 2) t2

ON t1.student\_id = t2.student\_id)

1. 运行结果如下图：



1. 查询学过“李平老师”所教的所有课的同学的学号、姓名；
2. 编写代码：

SELECT

student\_id '学号',

student\_name '姓名'

FROM

student

WHERE

student\_id IN (

SELECT student\_id

FROM score sc INNER JOIN

(SELECT subject\_id,teacher\_name

FROM `subject` sub,teacher tea

WHERE sub.teacher\_id = tea.teacher\_id) t1

ON sc.subject\_id = t1.subject\_id

WHERE

t1.teacher\_name = '李平老师'

GROUP BY student\_id

HAVING

COUNT(t1.teacher\_name = '李平老师') =

(SELECT COUNT(\*)

FROM `subject`

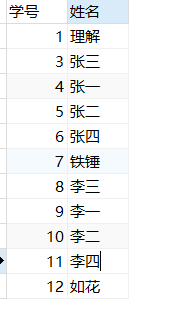
WHERE teacher\_id IN

(SELECT teacher\_id

FROM teacher

WHERE teacher\_name = '李平老师')))

1. 运行结果如下图：



1. 查询没有学全所有课的同学的学号、姓名；
2. 编写代码：

SELECT DISTINCT

sc.student\_id '学号',

student\_name '姓名'

FROM

student stu

INNER JOIN

score sc

ON

stu.student\_id = sc.student\_id

WHERE

sc.student\_id NOT IN(

SELECT

student\_id

FROM

score

GROUP BY student\_id

HAVING

COUNT(\*) = (

SELECT

COUNT(\*)

FROM

`subject`

)

)

1. 运行结果如下图：



1. 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；
2. 编写代码：

SELECT

student\_id '学号',

student\_name '姓名'

FROM

student

WHERE

student\_id IN (

SELECT

student\_id

FROM

score

WHERE

subject\_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(\*) = (

SELECT

COUNT(\*)

FROM

score

WHERE

student\_id = 2

)

)

1. 运行结果如下图：



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；

（1）编写代码：

SELECT

tb1.student\_id '学生ID',

tb2.score1 '生物',

tb3.score2 '物理',

tb4.score3 '体育',

tb5.score4 '美术',

tb1.count\_sub '有效课程数',

tb1.avg\_sc '有效平均分'

FROM

student stu

LEFT JOIN

(SELECT

stu.student\_id,

t1.count\_sub,

t1.avg\_sc

FROM

student stu

LEFT JOIN (

SELECT

student\_id,

COUNT(\*) count\_sub,

avg(score) avg\_sc

FROM

score

GROUP BY student\_id) t1

ON

stu.student\_id = t1.student\_id

) tb1

ON

stu.student\_id = tb1.student\_id

LEFT JOIN

(SELECT

stu.student\_id,

sc1.score1

FROM

student stu

LEFT JOIN (

SELECT

student\_id,

score score1

FROM

score

WHERE

subject\_id = 1) sc1

ON

stu.student\_id = sc1.student\_id

) tb2

ON

stu.student\_id = tb2.student\_id

LEFT JOIN

(SELECT

stu.student\_id,

sc2.score2

FROM

student stu

LEFT JOIN (

SELECT

student\_id,

score score2

FROM

score

WHERE

subject\_id = 2) sc2

ON

stu.student\_id = sc2.student\_id

) tb3

ON

stu.student\_id = tb3.student\_id

LEFT JOIN

(SELECT

stu.student\_id,

sc3.score3

FROM

student stu

LEFT JOIN (

SELECT

student\_id,

score score3

FROM

score

WHERE

subject\_id = 3) sc3

ON

stu.student\_id = sc3.student\_id

) tb4

ON

stu.student\_id = tb4.student\_id

LEFT JOIN

(SELECT

stu.student\_id,

sc4.score4

FROM

student stu

LEFT JOIN (

SELECT

student\_id,

score score4

FROM

score

WHERE

subject\_id = 4) sc4

ON

stu.student\_id = sc4.student\_id

) tb5

ON

stu.student\_id = tb5.student\_id

（2）运行结果如下图：



1. 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；
2. 编写代码：

SELECT

subject\_id '课程ID',

max(score) '最高分',

min(score) '最低分'

FROM

score

GROUP BY subject\_id

1. 运行结果如下图：



1. 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；
2. 编写代码：

SELECT

t1.subject\_id '课程编号',

t1.avg\_sc '平均成绩',

t2.passper '及格率'

FROM (

SELECT

subject\_id,

avg(score) avg\_sc

FROM

score

GROUP BY

subject\_id) t1

LEFT JOIN

(SELECT

pass\_cnt.subject\_id,

pass\_cnt.passcount/all\_cnt.allcount passper

FROM

(SELECT

subject\_id,

COUNT(\*) passcount

FROM

score

WHERE

score > 60

GROUP BY subject\_id) pass\_cnt

LEFT JOIN

(SELECT

subject\_id,

COUNT(\*) allcount

FROM

score

GROUP BY subject\_id) all\_cnt

ON

pass\_cnt.subject\_id = all\_cnt.subject\_id) t2

ON

t1.subject\_id = t2.subject\_id

ORDER BY

avg\_sc ASC,

passper DESC

1. 运行结果如下图：



1. 查询各科成绩前三名的记录:(不考虑成绩并列情况)
2. 编写代码：

select

t1.student\_id,t1.subject\_id,t1.score from score t1

left join

(

select score\_id,subject\_id,

(select score from score as s2 where s2.subject\_id = s1.subject\_id order by score desc limit 0, 1) as first\_num,

(select score from score as s2 where s2.subject\_id = s1.subject\_id order by score desc limit 1, 1) as second\_num

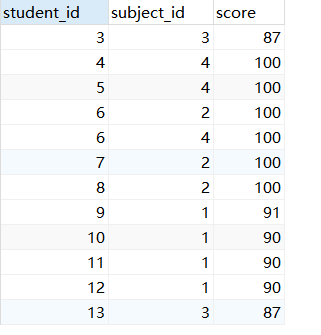
from score as s1

) t2

on t1.score\_id = t2.score\_id

where t1.score = t2.first\_num or t1.score = t2.second\_num ;

1. 运行结果如下图：



24.查询每门课程被选修的学生数；

（1）编写代码：

SELECT

subject\_id '课程ID',

COUNT(\*) '学生数'

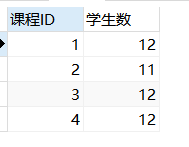
FROM

score

GROUP BY

subject\_id

（2）运行结果如下图：



1. 查询同名同姓学生名单，并统计同名人数；
2. 编写代码：

SELECT

student\_name '姓名',

t1.cnt '个数'

FROM(

SELECT

student\_name,

COUNT(\*) cnt

FROM

student

GROUP BY

student\_name) t1

WHERE

t1.cnt >= 2

1. 运行结果如下图：



1. 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；
2. 编写代码：

SELECT

subject\_id '课程ID',

avg(score) '平均成绩'

FROM

score

GROUP BY

subject\_id

ORDER BY

score ASC,

subject\_id DESC

1. 运行结果如下图：



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；

1. 编写代码：

SELECT

sc.student\_id '学号',

student\_name '姓名',

avg(score) '平均成绩'

FROM

student stu

INNER JOIN

score sc

ON

stu.student\_id = sc.student\_id

GROUP BY

sc.student\_id

HAVING

avg(score) > 85

1. 运行结果如下图：



1. 查询课程名称为“生物”，且分数低于60的学生姓名和分数；
2. 编写代码：

SELECT

subject\_name '课程名称',

student\_name '学生姓名',

score '分数'

FROM

student stu

INNER JOIN

score sc

ON

stu.student\_id = sc.student\_id

INNER JOIN

`subject` sub

ON

sc.subject\_id = sub.subject\_id

WHERE

sub.subject\_name = '生物'

AND

sc.score < 60

1. 运行结果如下图：



1. 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；
2. 编写代码：

SELECT

student\_id '学号',

student\_name '姓名'

FROM

student

WHERE

student\_id IN (

SELECT

student\_id

FROM

score

WHERE

subject\_id = 3

AND

score > 80

)

1. 运行结果如下图：



1. 求选了课程的学生人数
2. 编写代码：

SELECT

COUNT(\*)'选课人数'

FROM

(

SELECT DISTINCT

student\_id

FROM

score

) t1

1. 运行结果如下图：



1. 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；
2. 编写代码：

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

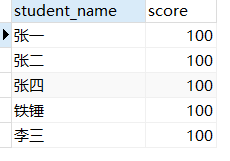
left join

student

on

t1.student\_id = student.student\_id

1. 运行结果如下图：



32.查询各个课程及相应的选修人数；

（1）编写代码：

SELECT

subject\_id '课程ID',

COUNT(\*) '选修人数'

FROM

score

GROUP BY

subject\_id

（2）运行结果如下图：



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；

34.查询每门课程成绩最好的前两名；

35.检索至少选修两门课程的学生学号；

（1）编写代码：

SELECT

student\_id '学号'

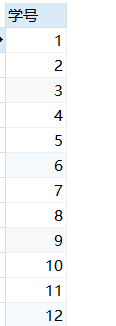
FROM

score

GROUP BY student\_id

HAVING COUNT(\*) >= 2

（2）运行结果如下图：



1. 查询全部学生都选修的课程的课程号和课程名；
2. 编写代码：

SELECT

sc.subject\_id '课程号',

subject\_name '课程名'

FROM

score sc

LEFT JOIN

`subject` sub

ON sc.subject\_id = sub.subject\_id

GROUP BY sc.subject\_id

HAVING

COUNT(\*) =

(

SELECT

COUNT(\*)

FROM

student

)

1. 运行结果如下图：



1. 查询没学过“李平”老师讲授的任一门课程的学生姓名；
2. 编写代码：

SELECT

student\_name '姓名'

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

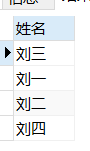
WHERE

teacher\_name = '李平老师'

)

))

1. 运行结果如下图：



1. 查询两门以上不及格课程的同学的学号及其平均成绩；
2. 编写代码：

select

student\_id '学号',

avg(score) '平均成绩'

from score

where score<60

group by student\_id

having count(\*)>=2;

1. 运行结果如下图：



1. 检索“004”课程分数小于60，按分数降序排列的同学学号；
2. 编写代码：

SELECT

student\_id '学号'

FROM

score

WHERE

score < 60

AND

subject\_id = 4

ORDER BY score DESC

（2）运行结果如下图：



40.删除“002”同学的“001”课程的成绩；

(1)编写代码：

delete from score where subject\_id = 1 and student\_id = 2