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.查询姓“张”的学生名单；

3.课程平均分从高到低显示

4.查询有课程成绩小于60分的同学的学号、姓名；

5.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；

6.查询出只选修了一门课程的全部学生的学号和姓名；

7.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；

8.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；

9.查询“生物”课程比“物理”课程成绩高的所有学生的学号；

10.查询平均成绩大于60分的同学的学号和平均成绩;

11.查询所有同学的学号、姓名、选课数、总成绩；

12.查询姓“李”的老师的个数；

13.查询没学过“张磊老师”课的同学的学号、姓名；

14.查询学过“1”并且也学过编号“2”课程的同学的学号、姓名；

15.查询学过“李平老师”所教的所有课的同学的学号、姓名；

16.查询没有学全所有课的同学的学号、姓名；

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；

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；

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；

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)

24.查询每门课程被选修的学生数；

25.查询同名同姓学生名单，并统计同名人数；

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；

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；

28.查询课程名称为“数学”，且分数低于60的学生姓名和分数；

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；

30.求选了课程的学生人数

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；

32.查询各个课程及相应的选修人数；

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；

36.查询全部学生都选修的课程的课程号和课程名；

37.查询没学过“叶平”老师讲授的任一门课程的学生姓名；

38.查询两门以上不及格课程的同学的学号及其平均成绩；

39.检索“004”课程分数小于60，按分数降序排列的同学学号；

40.删除“002”同学的“001”课程的成绩；

CREATE TABLE class(

class\_id INT(10) AUTO\_INCREMENT NOT NULL,

class\_name VARCHAR(64) not null,

primary key (class\_id)

);

alter table class convert to character set utf8;

INSERT INTO class (class\_id,class\_name)

VALUES ('1','三年二班'),

('2', '三年三班'),

('3', '一年二班'),

('4', '二年九班');

CREATE TABLE subject(

subject\_id INT(10) AUTO\_INCREMENT NOT NULL,

subject\_name VARCHAR(64) not null,

teacher\_id INT(10),

primary key (subject\_id)

);

alter table subject convert to character set utf8;

INSERT INTO subject(subject\_id,subject\_name,teacher\_id)

VALUES ('1', '生物', '1'),

('2', '物理', '2'),

('3', '体育', '3'),

('4', '美术', '2');

CREATE TABLE score\_student(

score\_id INT(10) AUTO\_INCREMENT NOT NULL,

subject\_id VARCHAR(64) not null,

student\_id INT(10),

score int(10),

primary key (score\_id)

);

alter table score\_student convert to character set utf8;

INSERT INTO score\_student(score\_id,student\_id,subject\_id,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'); .

CREATE TABLE student\_a(

student\_id INT(10) AUTO\_INCREMENT NOT NULL,

sex VARCHAR(64) not null,

class\_id INT(10),

student\_name VARCHAR(64),

primary key (student\_id)

);

alter table student\_a convert to character set utf8;

insert INTO student\_a (student\_id,sex,class\_id,student\_name)

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');

CREATE TABLE teacher(

teacher\_id INT(10) AUTO\_INCREMENT NOT NULL,

teacher\_name VARCHAR(64) not null,

primary key (teacher\_id)

);

alter table teacher convert to character set utf8;

INSERT INTO teacher (teacher\_id,teacher\_name)

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

select sex,COUNT(sex)

FROM student\_a

GROUP BY sex

-- 2

select student\_name

from student\_a

where student\_name LIKE '张%'

-- 3

select score\_id,AVG(score)

FROM score\_student

GROUP BY(subject\_id)

ORDER BY AVG(score) DESC

-- 4

select student\_a.student\_id,student\_a.student\_name,score\_student.score

FROM student\_a

INNER JOIN score\_student on student\_a.student\_id=score\_student.student\_id

WHERE score\_student.score<60

-- 5

SELECT \* FROM student\_a,

(SELECT DISTINCT score\_student.student\_id FROM score\_student WHERE subject\_id IN (SELECT subject\_id FROM score\_student WHERE student\_id = 1))b

WHERE student\_a.student\_id = b.student\_id and student\_a.student\_id!=1

-- 6

select \*

from Score\_student

group by student\_id

having count(\*) = 1

-- 7

select subject.subject\_name,score\_student.score\_id,max(score\_student.score),MIN(score\_student.score)

from score\_student

INNER JOIN subject on subject.subject\_id=score\_student.subject\_id

GROUP BY subject.subject\_id

-- 8

select d.student\_id,d.Student\_NAME

FROM student\_a d,

(select a.stu

FROM (select student\_id stu,score as 'score02'

FROM score\_student

where subject\_id=2) a,

(select student\_id stu,score as 'score01'

FROM score\_student

where subject\_id=1) b

where a.stu=b.stu and a.score02<b.score01)c

where d.student\_id=c.stu

-- 9

select d.student\_id,d.Student\_NAME

FROM student\_a d,

(select a.stu

FROM (select student\_id stu,score as 'score02'

FROM score\_student

where subject\_id=((select subject\_id

FROM subject

where subject\_name='生物'))) a,

(select student\_id stu,score as 'score01'

FROM score\_student

where subject\_id=((select subject\_id

FROM subject

where subject\_name='物理'))) b

where a.stu=b.stu and a.score02>b.score01)c

where d.student\_id=c.stu

-- 10

SELECT a.student\_id,a.score

FROM (SELECT student\_id,AVG(score) as'score'

FROM score\_student

GROUP BY student\_id) a

WHERE a.score>60

-- 11

SELECT tb2.student\_name,tb1.student\_id,tb1.count,tb1.sum

FROM student\_a tb2

LEFT JOIN

(select student\_id,count(\*)as 'count',sum(score)as 'sum'

FROM score\_student

GROUP BY student\_id) tb1

on tb2.student\_id=tb1.student\_id

-- 12

SELECT COUNT(\*)

FROM teacher

where teacher.teacher\_name like '李%'

-- 13

select tb1.student\_id,tb2.student\_name

FROM (SELECT \*

FROM score\_student

GROUP BY student\_id) tb1,student\_a tb2

WHERE tb2.student\_id=tb1.student\_id and tb1.subject\_id!=(select subject\_id

FROM subject

where teacher\_id=

(SELECT teacher\_id

FROM teacher

where teacher\_name='张磊老师'))

-- 14

select d.student\_id,d.Student\_NAME

FROM student\_a d,

(select a.stu

FROM (select student\_id stu,score as 'score02'

FROM score\_student

where subject\_id=((select subject\_id

FROM subject

where subject\_id=1))) a,

(select student\_id stu,score as 'score01'

FROM score\_student

where subject\_id=((select subject\_id

FROM subject

where subject\_id=2))) b

where a.stu=b.stu )c

where d.student\_id=c.stu

-- 15

select stu.student\_id,stu.student\_name

FROM student\_a stu

INNER JOIN

(select a.student\_id

FROM score\_student a

INNER JOIN (

select tb2.subject\_id

FROM teacher tb1

LEFT JOIN subject tb2

on tb1.teacher\_id=tb2.teacher\_id

where tb1.teacher\_name='李平老师') b

on b.subject\_id=a.subject\_id

GROUP BY a.student\_id) stuid

on stuid.student\_id=stu.student\_id

-- 16

select tb2.student\_id,tb2.student\_name,tb1.count as '课数'

FROM student\_a tb2

INNER JOIN

(select student\_id,COUNT(\*)as count

FROM score\_student

GROUP BY student\_id) tb1

on tb1.student\_id=tb2.student\_id

where tb1.count<(select COUNT(\*)FROM subject)

-- 17

select tb1.student\_id,tb1.student\_name

FROM student\_a tb1

INNER JOIN

(select student\_id,COUNT(\*)

FROM score\_student

where subject\_id in(select subject\_id

from score\_student

where student\_id=2)

GROUP BY student\_id HAVING COUNT(\*)=3) tb2

on tb2.student\_id=tb1.student\_id

select student\_id

from score\_student

where student\_id

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

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

group by student\_id

having count(\*)=(select count(\*)

from score\_student where student\_id=2) and student\_id !=2

-- 18

select \*

FROM score\_student

where subject\_id in

(select subject\_id

FROM subject

where teacher\_id=(select teacher\_id

FROM teacher

where teacher\_name='李平老师'))

SELECT student\_id

from student\_a

where student\_id

not in(select student\_id

FROM score\_student

where subject\_id=2)

-- 19

insert into score\_student(student\_id, subject\_id, score)

select student\_id,2,(select avg(score) from score\_student where subject\_id = 2)

from student\_a where student\_id not in (

select student\_id from score\_student where subject\_id = 2

)

-- 20

SELECT \*

FROM student\_a tb2

INNER JOIN (select student\_id,score as'c'

FROM score\_student

where subject\_id=3) tb1

on tb2.student\_id=tb1.student\_id

select sc.student\_id,

(select score from score\_student left join subject on score\_student.subject\_id = subject.subject\_id where subject.subject\_name = "生物" and score\_student.student\_id=sc.student\_id) as s,

(select score from score\_student left join subject on score\_student.subject\_id = subject.subject\_id where subject.subject\_name = "物理" and score\_student.student\_id=sc.student\_id) as w,

(select score from score\_student left join subject on score\_student.subject\_id = subject.subject\_id where subject.subject\_name = "体育" and score\_student.student\_id=sc.student\_id) as t,

count(sc.student\_id),

avg(sc.score)

from score\_student as sc

group by student\_id

ORDER BY avg(sc.score) desc

-- 21

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

FROM score\_student

GROUP BY subject\_id

-- 22

select subject\_id,AVG(score),sum(case when score\_student.score > 60 then 1 else 0 END)/count(1)\*100 as percent

FROM score\_student

GROUP BY subject\_id

ORDER BY avg(score) DESC

-- 23!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

SELECT

a.student\_id,

a.subject\_id,

a.score

FROM

score\_student a

LEFT JOIN score\_student b ON a.subject\_id = b.subject\_id

AND a.score <= b.score

GROUP BY

a.student\_id,

a.subject\_id,

a.score

HAVING

COUNT( b.student\_id ) <= 3

ORDER BY

a.subject\_id,

a.score DESC

-- 第2写法

SELECT

sub.subject\_name,

( SELECT score FROM score\_student sc2 WHERE sc2.subject\_id=sc.subject\_id ORDER BY score DESC LIMIT 0, 1 ) frist,

( SELECT score FROM score\_student sc2 WHERE sc2.subject\_id=sc.subject\_id ORDER BY score DESC LIMIT 1, 1 ) SECOND,

( SELECT score FROM score\_student sc2 WHERE sc2.subject\_id=sc.subject\_id ORDER BY score DESC LIMIT 2, 1 ) third

FROM

score\_student sc

INNER JOIN `subject` sub ON sub.subject\_id = sc.subject\_id

AND sub.subject\_id IN ( SELECT subject\_id FROM `subject` )

GROUP BY sc.subject\_id

ORDER BY

sc.subject\_id

-- 24

select tb1.subject\_id,tb2.subject\_name, count(\*)

from score\_student as tb1

INNER JOIN `subject` as tb2

on tb1.subject\_id=tb2.subject\_id

group by tb1.subject\_id;

-- 25

select student\_name,COUNT(\*)

FROM student\_a

GROUP BY student\_name HAVING COUNT(\*)>1

-- 26

select subject\_id,AVG(score)

FROM score\_student

GROUP BY subject\_id

ORDER BY avg(score) ,subject\_id

-- 27

select score\_student.student\_id,student\_a.student\_name,AVG(score) as avg

FROM score\_student

INNER JOIN student\_a

on student\_a.student\_id=score\_student.student\_id

GROUP BY student\_id HAVING avg>80

ORDER BY avg(score)

-- 28

select sc.student\_id,stu.student\_name

from score\_student as sc

INNER JOIN student\_a as stu

on stu.student\_id=sc.score

where subject\_id=(select subject\_id

FROM subject

where subject\_name ='物理') and score<60

-- 29

SELECT stu.student\_id,stu.student\_name

FROM score\_student sc

LEFT JOIN student\_a as stu

on stu.student\_id=sc.student\_id

where subject\_id=3 and score>80

-- 30

select COUNT(\*)

FROM

(select student\_id,COUNT(\*)

FROM score\_student

GROUP BY student\_id) as tb1

-- 31

select student\_name,max(score)

FROM score\_student

INNER JOIN student\_a

on student\_a.student\_id=score\_student.student\_id

where subject\_id=

(select sub.subject\_id

FROM teacher as th

INNER JOIN `subject` as sub

on sub.teacher\_id=th.teacher\_id

where th.teacher\_name='张磊老师')

-- 32

select `subject`.subject\_name,COUNT(\*)

FROM score\_student

INNER JOIN `subject`

on `subject`.subject\_id=score\_student.subject\_id

GROUP BY score\_student.subject\_id

-- 33

select t2.student\_id,t2.subject\_id,t1.student\_id,t1.subject\_id,t2.score

FROM score\_student as t2

INNER JOIN score\_student as t1

on t2.score=t1.score and t2.subject\_id != t1.subject\_id

-- 34

SELECT

sub.subject\_name,

( SELECT score FROM score\_student sc2 WHERE sc2.subject\_id=sc.subject\_id ORDER BY score DESC LIMIT 0, 1 ) frist,

( SELECT score FROM score\_student sc2 WHERE sc2.subject\_id=sc.subject\_id ORDER BY score DESC LIMIT 1, 1 ) SECOND

FROM

score\_student sc

INNER JOIN `subject` sub ON sub.subject\_id = sc.subject\_id

AND sub.subject\_id IN ( SELECT subject\_id FROM `subject` )

GROUP BY sc.subject\_id

ORDER BY

sc.subject\_id

-- 35

select \*,COUNT(\*)

FROM score\_student

GROUP BY student\_id HAVING COUNT(\*)>1

-- 36

SELECT sc.subject\_id,sub.subject\_name

FROM score\_student as sc

INNER JOIN `subject` as sub

on sc.subject\_id=sub.subject\_id

GROUP BY subject\_id HAVING COUNT(\*)=(select COUNT(\*)

FROM student\_a)

-- 37

select st.student\_name

FROM score\_student stu

INNER JOIN student\_a st

on st.student\_id=stu.student\_id

where subject\_id not in(

select sub.subject\_id

FROM teacher th

INNER JOIN `subject` sub

on th.teacher\_id=sub.teacher\_id

where th.teacher\_name='李平老师')

GROUP BY st.student\_name

-- 38

select \*,COUNT(\*),avg(score)

FROM score\_student

where score<60

GROUP BY student\_id HAVING COUNT(\*)>1

-- 39

SELECT \*

FROM score\_student

where subject\_id=4 and score<60

ORDER BY DESC

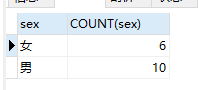
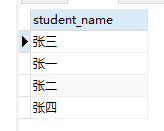
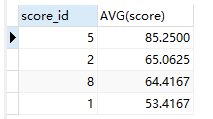
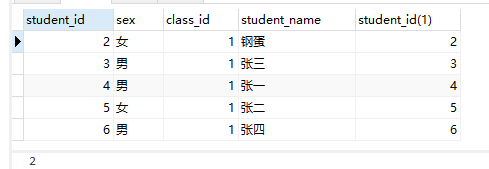
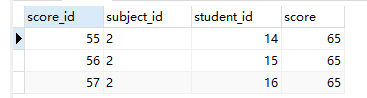
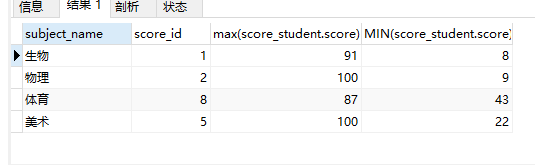
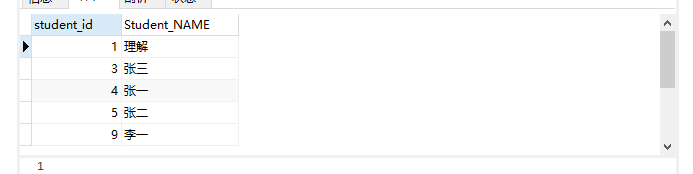
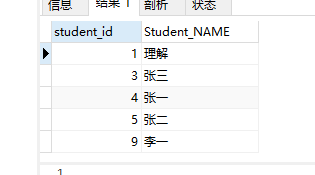
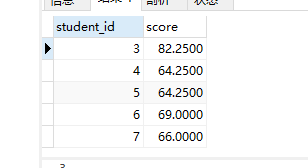
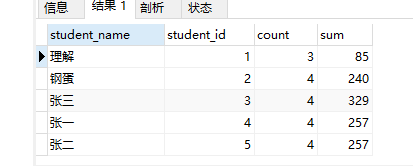
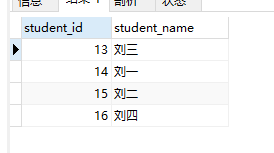
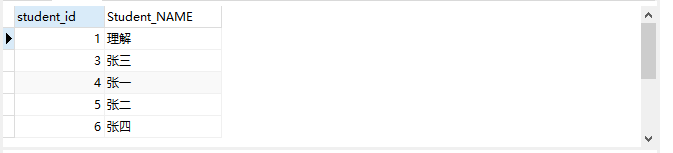
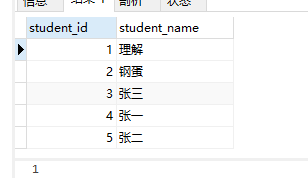
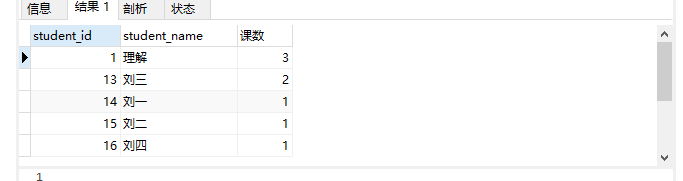
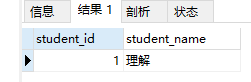
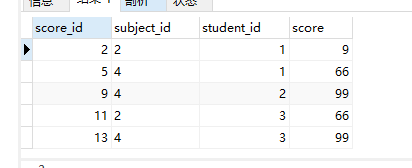
-- 40

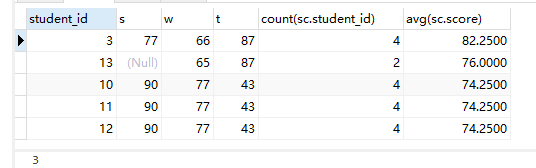
-- DELETE

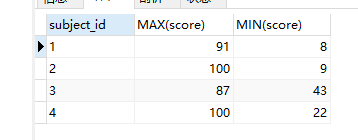
select \*

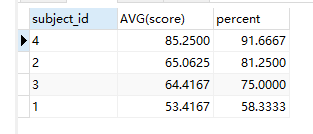
FROM score\_student

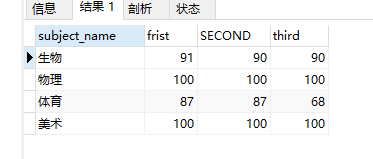
where student\_id=2 and subject\_id=1

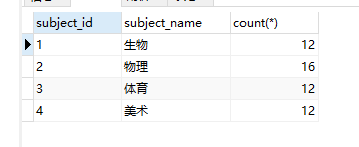
1. 
2. 
3. 
4. 
5. 
6. 
7. 
8. 
9. 
10. 
11. 
12. 
13. 
14. 
15. 
16. 
17. 
18. 

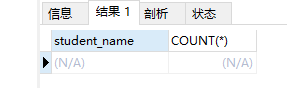
20. 

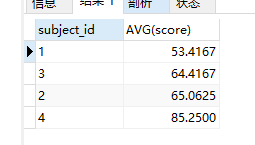
21. 

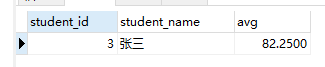
22. 

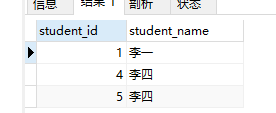
23. 

24. 

25. 

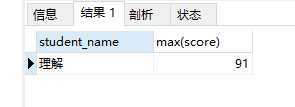
26. 

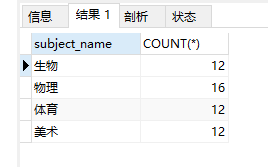
27. 

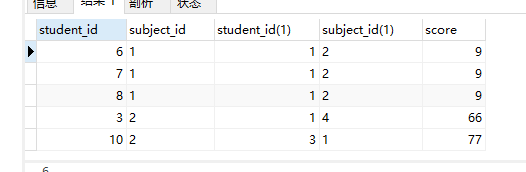
28. 

29. 

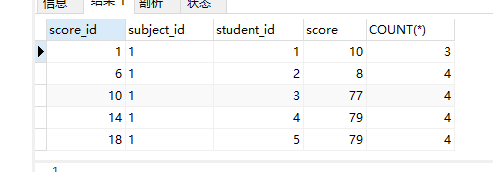
30 

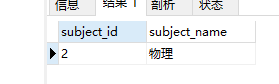
31. 

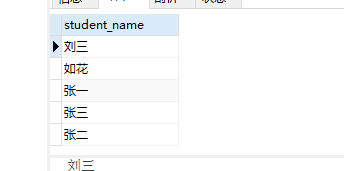
32 

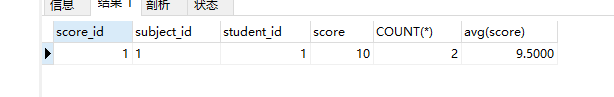
33 

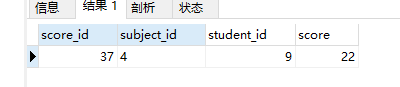
34 

35 

36 

37 

38 

39. 

40 