表名和字段

–1.学生表

Student(s\_id,s\_name,s\_birth,s\_sex) --学生编号,学生姓名, 出生年月,学生性别

–2.课程表

Course(c\_id,c\_name,t\_id) – --课程编号, 课程名称, 教师编号

–3.教师表

Teacher(t\_id,t\_name) --教师编号,教师姓名

–4.成绩表

Score(s\_id,c\_id,s\_score) --学生编号,课程编号,分数

测试数据

--建表

--学生表

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`)

);

--课程表

CREATE TABLE `Course`(

`c\_id` VARCHAR(20),

`c\_name` VARCHAR(20) NOT NULL DEFAULT '',

`t\_id` VARCHAR(20) NOT NULL,

PRIMARY KEY(`c\_id`)

);

--教师表

CREATE TABLE `Teacher`(

`t\_id` VARCHAR(20),

`t\_name` VARCHAR(20) NOT NULL DEFAULT '',

PRIMARY KEY(`t\_id`)

);

--成绩表

CREATE TABLE `Score`(

`s\_id` VARCHAR(20),

`c\_id` VARCHAR(20),

`s\_score` INT(3),

PRIMARY KEY(`s\_id`,`c\_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');

--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');

--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

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

练习题和sql语句

-- 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 a.\* ,b.s\_score as 01\_score,c.s\_score as 02\_score from

student a

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

left join score c on a.s\_id=c.s\_id and c.c\_id='02' or c.c\_id = NULL where b.s\_score>c.s\_score

--也可以这样写

select a.\*,b.s\_score as 01\_score,c.s\_score as 02\_score 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'

and b.s\_score>c.s\_score

-- 2、查询"01"课程比"02"课程成绩低的学生的信息及课程分数

select a.\* ,b.s\_score as 01\_score,c.s\_score as 02\_score from

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

join score c on a.s\_id=c.s\_id and c.c\_id='02' where b.s\_score<c.s\_score

-- 3、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

select b.s\_id,b.s\_name,ROUND(AVG(a.s\_score),2) as avg\_score from

student b

join score a on b.s\_id = a.s\_id

GROUP BY b.s\_id,b.s\_name HAVING avg\_score >=60;

-- 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)

select b.s\_id,b.s\_name,ROUND(AVG(a.s\_score),2) as avg\_score from

student b

left join score a on b.s\_id = a.s\_id

GROUP BY b.s\_id,b.s\_name HAVING avg\_score <60

union

select a.s\_id,a.s\_name,0 as avg\_score from

student a

where a.s\_id not in (

select distinct s\_id from score);

-- 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 a.s\_id,a.s\_name,count(b.c\_id) as sum\_course,sum(b.s\_score) as sum\_score from

student a

left join score b on a.s\_id=b.s\_id

GROUP BY a.s\_id,a.s\_name;

-- 6、查询"李"姓老师的数量

select count(t\_id) from teacher where t\_name like '李%';

-- 7、查询学过"张三"老师授课的同学的信息

select a.\* from

student a

join score b on a.s\_id=b.s\_id where b.c\_id in(

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

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

-- 8、查询没学过"张三"老师授课的同学的信息

select \* from

student c

where c.s\_id not in(

select a.s\_id from student a join score b on a.s\_id=b.s\_id where b.c\_id in(

select a.c\_id from course a join teacher b on a.t\_id = b.t\_id where t\_name ='张三'));

-- 9、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';

-- 10、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 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')

-- 11、查询没有学全所有课程的同学的信息

--@wendiepei的写法

select s.\* from student s

left join Score s1 on s1.s\_id=s.s\_id

group by s.s\_id having count(s1.c\_id)<(select count(\*) from course)

--@k1051785839的写法

select \*

from student

where s\_id not in(

select s\_id from score t1

group by s\_id having count(\*) =(select count(distinct c\_id) from course))

-- 12、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 \* 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')

);

-- 13、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

--@ouyang\_1993的写法

SELECT

Student.\*

FROM

Student

WHERE

s\_id IN (SELECT s\_id FROM Score GROUP BY s\_id HAVING COUNT(s\_id) = (

#下面的语句是找到'01'同学学习的课程数

SELECT COUNT(c\_id) FROM Score WHERE s\_id = '01'

)

)

AND s\_id NOT IN (

#下面的语句是找到学过‘01’同学没学过的课程，有哪些同学。并排除他们

SELECT s\_id FROM Score

WHERE c\_id IN(

#下面的语句是找到‘01’同学没学过的课程

SELECT DISTINCT c\_id FROM Score

WHERE c\_id NOT IN (

#下面的语句是找出‘01’同学学习的课程

SELECT c\_id FROM Score WHERE s\_id = '01'

)

) GROUP BY s\_id

) #下面的条件是排除01同学

AND s\_id NOT IN ('01')

--@k1051785839的写法

SELECT

t3.\*

FROM

(

SELECT

s\_id,

group\_concat(c\_id ORDER BY c\_id) group1

FROM

score

WHERE

s\_id &lt;> '01'

GROUP BY

s\_id

) t1

INNER JOIN (

SELECT

group\_concat(c\_id ORDER BY c\_id) group2

FROM

score

WHERE

s\_id = '01'

GROUP BY

s\_id

) t2 ON t1.group1 = t2.group2

INNER JOIN student t3 ON t1.s\_id = t3.s\_id

-- 14、查询没学过"张三"老师讲授的任一门课程的学生姓名

select a.s\_name from student a where a.s\_id not in (

select s\_id from score where c\_id =

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

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

-- 15、查询两门及其以上不及格课程的同学的学号，姓名及其平均成绩

select a.s\_id,a.s\_name,ROUND(AVG(b.s\_score)) from

student a

left join score b on a.s\_id = b.s\_id

where a.s\_id in(

select s\_id from score where s\_score<60 GROUP BY s\_id having count(1)>=2)

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

-- 16、检索"01"课程分数小于60，按分数降序排列的学生信息

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;

-- 17、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 s\_score from score where s\_id=a.s\_id and c\_id='01') as 语文,

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

(select s\_score from score where s\_id=a.s\_id and c\_id='03') as 英语,

round(avg(s\_score),2) as 平均分 from score a GROUP BY a.s\_id ORDER BY 平均分 DESC;

--@喝完这杯还有一箱的写法

SELECT a.s\_id,MAX(CASE a.c\_id WHEN '01' THEN a.s\_score END ) 语文,

MAX(CASE a.c\_id WHEN '02' THEN a.s\_score END ) 数学,

MAX(CASE a.c\_id WHEN '03' THEN a.s\_score END ) 英语,

avg(a.s\_score),b.s\_name FROM Score a JOIN Student b ON a.s\_id=b.s\_id GROUP BY a.s\_id ORDER BY 5 DESC

-- 18.查询各科成绩最高分、最低分和平均分：以如下形式显示：课程ID，课程name，最高分，最低分，平均分，及格率，中等率，优良率，优秀率

--及格为>=60，中等为：70-80，优良为：80-90，优秀为：>=90

select a.c\_id,b.c\_name,MAX(s\_score),MIN(s\_score),ROUND(AVG(s\_score),2),

ROUND(100\*(SUM(case when a.s\_score>=60 then 1 else 0 end)/SUM(case when a.s\_score then 1 else 0 end)),2) as 及格率,

ROUND(100\*(SUM(case when a.s\_score>=70 and a.s\_score<=80 then 1 else 0 end)/SUM(case when a.s\_score then 1 else 0 end)),2) as 中等率,

ROUND(100\*(SUM(case when a.s\_score>=80 and a.s\_score<=90 then 1 else 0 end)/SUM(case when a.s\_score then 1 else 0 end)),2) as 优良率,

ROUND(100\*(SUM(case when a.s\_score>=90 then 1 else 0 end)/SUM(case when a.s\_score then 1 else 0 end)),2) as 优秀率

from score a left join course b on a.c\_id = b.c\_id GROUP BY a.c\_id,b.c\_name

-- 19、按各科成绩进行排序，并显示排名

-- mysql没有rank函数

select a.s\_id,a.c\_id,

@i:=@i +1 as i保留排名,

@k:=(case when @score=a.s\_score then @k else @i end) as rank不保留排名,

@score:=a.s\_score as score

from (

select s\_id,c\_id,s\_score from score GROUP BY s\_id,c\_id,s\_score ORDER BY s\_score DESC

)a,(select @k:=0,@i:=0,@score:=0)s

--@k1051785839的写法

(select \* from (select

t1.c\_id,

t1.s\_score,

(select count(distinct t2.s\_score) from score t2 where t2.s\_score>=t1.s\_score and t2.c\_id='01') rank

FROM score t1 where t1.c\_id='01'

order by t1.s\_score desc) t1)

union

(select \* from (select

t1.c\_id,

t1.s\_score,

(select count(distinct t2.s\_score) from score t2 where t2.s\_score>=t1.s\_score and t2.c\_id='02') rank

FROM score t1 where t1.c\_id='02'

order by t1.s\_score desc) t2)

union

(select \* from (select

t1.c\_id,

t1.s\_score,

(select count(distinct t2.s\_score) from score t2 where t2.s\_score>=t1.s\_score and t2.c\_id='03') rank

FROM score t1 where t1.c\_id='03'

order by t1.s\_score desc) t3)

-- 20、查询学生的总成绩并进行排名

select a.s\_id,

@i:=@i+1 as i,

@k:=(case when @score=a.sum\_score then @k else @i end) as rank,

@score:=a.sum\_score as score

from (select s\_id,SUM(s\_score) as sum\_score from score GROUP BY s\_id ORDER BY sum\_score DESC)a,

(select @k:=0,@i:=0,@score:=0)s

-- 21、查询不同老师所教不同课程平均分从高到低显示

select a.t\_id,c.t\_name,a.c\_id,ROUND(avg(s\_score),2) as avg\_score from course a

left join score b on a.c\_id=b.c\_id

left join teacher c on a.t\_id=c.t\_id

GROUP BY a.c\_id,a.t\_id,c.t\_name ORDER BY avg\_score DESC;

-- 22、查询所有课程的成绩第2名到第3名的学生信息及该课程成绩

select d.\*,c.排名,c.s\_score,c.c\_id from (

select a.s\_id,a.s\_score,a.c\_id,@i:=@i+1 as 排名 from score a,(select @i:=0)s where a.c\_id='01'

ORDER BY a.s\_score DESC

)c

left join student d on c.s\_id=d.s\_id

where 排名 BETWEEN 2 AND 3

UNION

select d.\*,c.排名,c.s\_score,c.c\_id from (

select a.s\_id,a.s\_score,a.c\_id,@j:=@j+1 as 排名 from score a,(select @j:=0)s where a.c\_id='02'

ORDER BY a.s\_score DESC

)c

left join student d on c.s\_id=d.s\_id

where 排名 BETWEEN 2 AND 3

UNION

select d.\*,c.排名,c.s\_score,c.c\_id from (

select a.s\_id,a.s\_score,a.c\_id,@k:=@k+1 as 排名 from score a,(select @k:=0)s where a.c\_id='03'

ORDER BY a.s\_score DESC

)c

left join student d on c.s\_id=d.s\_id

where 排名 BETWEEN 2 AND 3;

-- 23、统计各科成绩各分数段人数：课程编号,课程名称,[100-85],[85-70],[70-60],[0-60]及所占百分比

select distinct f.c\_name,a.c\_id,b.`85-100`,b.百分比,c.`70-85`,c.百分比,d.`60-70`,d.百分比,e.`0-60`,e.百分比 from score a

left join (select c\_id,SUM(case when s\_score >85 and s\_score <=100 then 1 else 0 end) as `85-100`,

ROUND(100\*(SUM(case when s\_score >85 and s\_score <=100 then 1 else 0 end)/count(\*)),2) as 百分比

from score GROUP BY c\_id)b on a.c\_id=b.c\_id

left join (select c\_id,SUM(case when s\_score >70 and s\_score <=85 then 1 else 0 end) as `70-85`,

ROUND(100\*(SUM(case when s\_score >70 and s\_score <=85 then 1 else 0 end)/count(\*)),2) as 百分比

from score GROUP BY c\_id)c on a.c\_id=c.c\_id

left join (select c\_id,SUM(case when s\_score >60 and s\_score <=70 then 1 else 0 end) as `60-70`,

ROUND(100\*(SUM(case when s\_score >60 and s\_score <=70 then 1 else 0 end)/count(\*)),2) as 百分比

from score GROUP BY c\_id)d on a.c\_id=d.c\_id

left join (select c\_id,SUM(case when s\_score >=0 and s\_score <=60 then 1 else 0 end) as `0-60`,

ROUND(100\*(SUM(case when s\_score >=0 and s\_score <=60 then 1 else 0 end)/count(\*)),2) as 百分比

from score GROUP BY c\_id)e on a.c\_id=e.c\_id

left join course f on a.c\_id = f.c\_id

-- 24、查询学生平均成绩及其名次

select a.s\_id,

@i:=@i+1 as '不保留空缺排名',

@k:=(case when @avg\_score=a.avg\_s then @k else @i end) as '保留空缺排名',

@avg\_score:=avg\_s as '平均分'

from (select s\_id,ROUND(AVG(s\_score),2) as avg\_s from score GROUP BY s\_id ORDER BY avg\_s DESC)a,(select @avg\_score:=0,@i:=0,@k:=0)b;

-- 25、查询各科成绩前三名的记录

-- 1.选出b表比a表成绩大的所有组

-- 2.选出比当前id成绩大的 小于三个的

select a.s\_id,a.c\_id,a.s\_score from score a

left join score b on a.c\_id = b.c\_id and a.s\_score<b.s\_score

group by a.s\_id,a.c\_id,a.s\_score HAVING COUNT(b.s\_id)<3

ORDER BY a.c\_id,a.s\_score DESC

-- 26、查询每门课程被选修的学生数

select c\_id,count(s\_id) from score a GROUP BY c\_id

-- 27、查询出只有两门课程的全部学生的学号和姓名

select s\_id,s\_name from student where s\_id in(

select s\_id from score GROUP BY s\_id HAVING COUNT(c\_id)=2);

-- 28、查询男生、女生人数

select s\_sex,COUNT(s\_sex) as 人数 from student GROUP BY s\_sex

-- 29、查询名字中含有"风"字的学生信息

select \* from student where s\_name like '%风%';

-- 30、查询同名同性学生名单，并统计同名人数

select a.s\_name,a.s\_sex,count(\*) from student a JOIN

student b on a.s\_id !=b.s\_id and a.s\_name = b.s\_name and a.s\_sex = b.s\_sex

GROUP BY a.s\_name,a.s\_sex

-- 31、查询1990年出生的学生名单

select s\_name from student where s\_birth like '1990%'

-- 32、查询每门课程的平均成绩，结果按平均成绩降序排列，平均成绩相同时，按课程编号升序排列

select c\_id,ROUND(AVG(s\_score),2) as avg\_score from score GROUP BY c\_id ORDER BY avg\_score DESC,c\_id ASC

-- 33、查询平均成绩大于等于85的所有学生的学号、姓名和平均成绩

select a.s\_id,b.s\_name,ROUND(avg(a.s\_score),2) as avg\_score from score a

left join student b on a.s\_id=b.s\_id GROUP BY s\_id HAVING avg\_score>=85

-- 34、查询课程名称为"数学"，且分数低于60的学生姓名和分数

select a.s\_name,b.s\_score from score b join student a on a.s\_id=b.s\_id where b.c\_id=(

select c\_id from course where c\_name ='数学') and b.s\_score<60

-- 35、查询所有学生的课程及分数情况；

select a.s\_id,a.s\_name,

SUM(case c.c\_name when '语文' then b.s\_score else 0 end) as '语文',

SUM(case c.c\_name when '数学' then b.s\_score else 0 end) as '数学',

SUM(case c.c\_name when '英语' then b.s\_score else 0 end) as '英语',

SUM(b.s\_score) as '总分'

from student a left join score b on a.s\_id = b.s\_id

left join course c on b.c\_id = c.c\_id

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

-- 36、查询任何一门课程成绩在70分以上的姓名、课程名称和分数；

select a.s\_name,b.c\_name,c.s\_score from course b left join score c on b.c\_id = c.c\_id

left join student a on a.s\_id=c.s\_id where c.s\_score>=70

-- 37、查询不及格的课程

select a.s\_id,a.c\_id,b.c\_name,a.s\_score from score a left join course b on a.c\_id = b.c\_id

where a.s\_score<60

--38、查询课程编号为01且课程成绩在80分以上的学生的学号和姓名；

select a.s\_id,b.s\_name from score a LEFT JOIN student b on a.s\_id = b.s\_id

where a.c\_id = '01' and a.s\_score>80

-- 39、求每门课程的学生人数

select count(\*) from score GROUP BY c\_id;

-- 40、查询选修"张三"老师所授课程的学生中，成绩最高的学生信息及其成绩

-- 查询老师id

select c\_id from course c,teacher d where c.t\_id=d.t\_id and d.t\_name='张三'

-- 查询最高分（可能有相同分数）

select MAX(s\_score) from score where c\_id='02'

-- 查询信息

select a.\*,b.s\_score,b.c\_id,c.c\_name from student a

LEFT JOIN score b on a.s\_id = b.s\_id

LEFT JOIN course c on b.c\_id=c.c\_id

where b.c\_id =(select c\_id from course c,teacher d where c.t\_id=d.t\_id and d.t\_name='张三')

and b.s\_score in (select MAX(s\_score) from score where c\_id='02')

-- 41、查询不同课程成绩相同的学生的学生编号、课程编号、学生成绩

select DISTINCT b.s\_id,b.c\_id,b.s\_score from score a,score b where a.c\_id != b.c\_id and a.s\_score = b.s\_score

-- 42、查询每门功成绩最好的前两名

-- 牛逼的写法

select a.s\_id,a.c\_id,a.s\_score from score a

where (select COUNT(1) from score b where b.c\_id=a.c\_id and b.s\_score>=a.s\_score)<=2 ORDER BY a.c\_id

-- 43、统计每门课程的学生选修人数（超过5人的课程才统计）。要求输出课程号和选修人数，查询结果按人数降序排列，若人数相同，按课程号升序排列

select c\_id,count(\*) as total from score GROUP BY c\_id HAVING total>5 ORDER BY total,c\_id ASC

-- 44、检索至少选修两门课程的学生学号

select s\_id,count(\*) as sel from score GROUP BY s\_id HAVING sel>=2

-- 45、查询选修了全部课程的学生信息

select \* from student where s\_id in(

select s\_id from score GROUP BY s\_id HAVING count(\*)=(select count(\*) from course))

--46、查询各学生的年龄

-- 按照出生日期来算，当前月日 < 出生年月的月日则，年龄减一

select s\_birth,(DATE\_FORMAT(NOW(),'%Y')-DATE\_FORMAT(s\_birth,'%Y') -

(case when DATE\_FORMAT(NOW(),'%m%d')>DATE\_FORMAT(s\_birth,'%m%d') then 0 else 1 end)) as age

from student;

-- 47、查询本周过生日的学生

select \* from student where WEEK(DATE\_FORMAT(NOW(),'%Y%m%d'))=WEEK(s\_birth)

select \* from student where YEARWEEK(s\_birth)=YEARWEEK(DATE\_FORMAT(NOW(),'%Y%m%d'))

select WEEK(DATE\_FORMAT(NOW(),'%Y%m%d'))

-- 48、查询下周过生日的学生

select \* from student where WEEK(DATE\_FORMAT(NOW(),'%Y%m%d'))+1 =WEEK(s\_birth)

-- 49、查询本月过生日的学生

select \* from student where MONTH(DATE\_FORMAT(NOW(),'%Y%m%d')) =MONTH(s\_birth)

-- 50、查询下月过生日的学生

select \* from student where MONTH(DATE\_FORMAT(NOW(),'%Y%m%d'))+1 =MONTH(s\_birth)