

work-sql

余旺

April 2024

首先需要说明，对于下面的数据如果查询结果为空，都进行了插入或修改的处理，使其结果能够展示出来

开头的处理：加入本人 22301056 的 student 相关信息并插入自己设定的分数信息，对表中重复的 cid 和 grade 做了删除处理，如下图

WHERE ORDER BY Cid							
	Sno		Cid		Grade		
1	22309010		1		69		
2	22301056		1		80		
3	22309100		2		82		
4	22309015		2		79		
5	22301056		2		80		
6	22309012		3		88		
7	22309015		3		<null>		
8	22309015		3		<null>		
9	22301056		3		80		
10	22309030		3		78		
11	22309020		3		95		
12	22309100		4		67		
13	22309148		4		58		
14	22309020		4		88		
15	22309048		8		97		
16	22309075		8		91		
17	22309150		9		89		
18	22309160		9		71		

12	22309160	杨久玲	女	2005-12-12	RJ2206	0	22301056	18	<null>
13	22309010	李一在	男	2005-10-01	RJ2201	0	22301056	18	<null>
14	22309015	刘三晶	女	2005-05-22	RJ2201	0	22301056	18	<null>
15	22301056	余旺	男	2004-04-26	RJ2202	0	22301056	18	<null>

- 1 查询所有同学的学分情况 (假设课程成绩 ≥ 60 时可获得该门课程的学分), 显示学号、姓名、总学分 (用 JOIN)

```
46 ✓ select s.Sno, s.Sname, COALESCE(SUM(co.Ccredit), 0) AS TotalCredit
47 FROM student s
48 left JOIN sc ON s.Sno = sc.Sno AND sc.Grade >= 60
49 left JOIN teaching a ON a.cid=sc.Cid
50 left JOIN course co ON a.Cno = co.Cno
51 GROUP BY s.Sno, s.Sname;
```

	Sno	Sname	TotalCredit
1	22301056	余旺	12
2	22309010	李一在	4
3	22309012	葛畅	4
4	22309015	刘三晶	4
5	22309020	杨四敏	8
6	22309030	胡五斌	4
7	22309048	赵陈毅泽	4
8	22309050	王威	0
9	22309067	赵六玮	0
10	22309075	王娜	4
11	22309088	秦琪键	0
12	22309100	田仪	8
13	22309148	赵八砚	0
14	22309150	杨青	4
15	22309160	杨久玲	4

2 查询所有同学的平均成绩及选课门数，显示学号、姓名、平均成绩、选课门数

```
-- (2) 查询所有同学的平均成绩及选课门数，显示学号、姓名、平均成绩、选课门数
SELECT s.Sno, s.Sname, AVG(sc.Grade) AS AvgGrade, COUNT(sc.Cid) AS CourseCount
FROM student s
left JOIN sc ON s.Sno = sc.Sno
GROUP BY s.Sno, s.Sname;
```

	Sno	Sname	AvgGrade	CourseCount
1	22301056	余旺	80.0000	3
2	22309010	李一在	69.0000	1
3	22309012	葛畅	88.0000	1
4	22309015	刘三晶	79.0000	3
5	22309020	杨四敏	91.5000	2
6	22309030	胡五斌	78.0000	1
7	22309048	赵陈毅泽	97.0000	1
8	22309050	王威	<null>	0
9	22309067	赵六玮	<null>	0
10	22309075	王娜	91.0000	1
11	22309088	秦琪键	<null>	0
12	22309100	田仪	74.5000	2
13	22309148	赵八砚	58.0000	1
	22309150	杨青	89.0000	1
15	22309160	杨久玲	71.0000	1

3 查询所有选修了课程但未参加考试的所有同学及相应的课程，显示学号、姓名、课程号、课程名称

```
-- (3) 查询所有选修了课程但未参加考试的所有同学及相应的课程，显示学号、姓名、课程号、课程名称
select s.Sno, s.Sname, co.Cno, co.Cname
from student s
join sc ON s.Sno = sc.Sno
join teaching a ON a.cid=sc.Cid
join course co ON a.Cno = co.Cno
where sc.Grade IS NULL;
```

	Sno	Sname	Cno	Cname
1	22309015	刘三晶	800003	数据结构
2	22309015	刘三晶	800003	数据结构

4 查询所有选修了课程但考试不及格的所有同学及相应的课程，显示学号、姓名、课程号、课程名称、成绩

```
-- (4) 查询所有选修了课程但考试不及格的所有同学及相应的课程，显示学号、姓名、课程号、课程名称、成绩
select s.Sno, s.Sname, co.Cno, co.Cname, sc.Grade
from student s
join sc ON s.Sno = sc.Sno
join teaching a ON a.cid=sc.Cid
join course co ON a.Cno = co.Cno
where sc.Grade <60;
```

Sno	Sname	Cno	Cname	Grade
22309148	赵八砚	810011	数据库系统	58

(5) 查询选修了课程名为“面向对象程序设计”的所有同学及成绩情况，显示学生姓名、课程成绩 (用 ANY 运算符)

5 查询选修了课程名为“面向对象程序设计”的所有同学及成绩情况，显示学生姓名、课程成绩 (用 ANY 运算符)

```
-- 这里我们知道每个学生sc表中的cid都能够唯一对应找到他是否选了面向对象程序设计
SELECT s.Sname, b.Grade
FROM student s,sc b
WHERE s.Sno = b.Sno and Cid = ANY (
    SELECT sc.Cid
    FROM sc
    JOIN teaching a ON a.cid = sc.Cid
    JOIN course co ON a.Cno = co.Cno
    WHERE co.Cname = '面向对象程序设计'
);
```

Sname	Grade
田仪	82
刘三晶	79
余旺	80

6 查询“软件开发”系的所有同学及成绩情况，显示学号、姓名、班级名称、课程号、课程名称、成绩

381

where co.Cname='面向对象程序设计';

382

-- (6) 查询“软件开发”系的所有同学及成绩情况，显示学号、姓名、班级名称、课程号、课程名称、成绩

383

select s.Sno, s.Sname, b.classname, co.Cno, co.Cname, sc.Grade

384

from student s

385

join class b on b.Classno=s.Classno

386

join sc ON s.Sno = sc.Sno

387

join teaching a ON a.cid=sc.Cid

388

join course co ON a.Cno = co.Cno

389

where b.Classdept='软件开发';

×

Sno

Sname

classname

Cno

Cname

Grade

1

22309010

李一在

软件2201

800001

程序设计基础

69

2

22309015

刘三晶

软件2201

800002

面向对象程序设

79

3

22309012

葛畅

软件2201

800003

数据结构

88

4

22309015

刘三晶

软件2201

800003

数据结构

<null>

5

22309015

刘三晶

软件2201

800003

数据结构

<null>

6

22309048

赵陈毅泽

软件2202

800003

数据结构

97

7

22309020

杨四敏

软件2201

810011

数据库系统

88

8

22301056

余旺

软件2202

800001

程序设计基础

80

9

22301056

余旺

软件2202

800002

面向对象程序设

80

10

22301056

余旺

软件2202

800003

数据结构

80

11

22309030

胡五斌

软件2201

800003

数据结构

78

12

22309020

杨四敏

软件2201

800003

数据结构

95

7 查询成绩低于同门课程平均成绩的信息，显示学生学号、姓名、课程名称及低于平均成绩的值（即比平均成绩低多少）

因为该题是在做完后面的题目的时候回过头来重新做，所以 sc 中的数据都已经和原来不一样了，所以给出了分数如下所示

WHERE				ORDER BY			
	Sno	Cid	Grade				
1	22309010	1	69				
2	22309100	2	86				
3	22309015	2	79				
4	22309012	3	88				
5	22309048	8	100				
6	22309075	8	96				
7	22309100	4	70				
8	22309148	4	58				
9	22309020	4	88				
10	22309150	9	93				
11	22309160	9	79				
12	22301056	1	80				
13	22301056	2	80				
14	22301056	3	80				
15	22309030	3	78				
16	22309020	3	95				
17	22301056	4	60				
18	22301056	5	60				
19	22301056	11	60				

```
430
431
432 ✓ SELECT s.Sno, s.Sname, co.Cname, sc.Grade - avg_grade.below_avg AS below_avg
433 FROM student s
434 JOIN sc ON s.Sno = sc.Sno
435 JOIN teaching t ON t.CID = sc.Cid
436 JOIN course co ON t.Cno = co.Cno
437 JOIN (
438     SELECT sc.Cid, AVG(sc.Grade) AS below_avg
439     FROM sc
440     GROUP BY sc.Cid
441 ) AS avg_grade ON sc.Cid = avg_grade.Cid
442 WHERE sc.Grade - avg_grade.below_avg > 0;
```

×

	Sno	Sname	Cname	below_avg
1	22309100	田仪	面向对象程序设计	4.3333
2	22309012	葛畅	数据结构	2.7500
3	22309048	赵陈毅泽	数据结构	2.0000
4	22309100	田仪	数据库系统	1.0000
5	22309020	杨四敏	数据库系统	19.0000
6	22309150	杨青	数据库系统	7.0000
7	22301056	余旺	程序设计基础	5.5000
8	22309020	杨四敏	数据结构	9.7500

443

444

8 查询和“葛畅”在同一班级的同学的姓名（使用子查询）

430											
431	-- (8) 查询和“葛畅”在同一班级的同学的姓名（使用子查询）										
432 ✓	SELECT Sname										
433	FROM student										
434	WHERE Classno IN (SELECT Classno FROM student WHERE Sname = '葛畅') and Sname != '葛畅';										
×	<table> <tr> <th></th><th>Sname</th></tr> <tr> <td>1</td><td>李一在</td></tr> <tr> <td>2</td><td>刘三晶</td></tr> <tr> <td>3</td><td>杨四敏</td></tr> <tr> <td>4</td><td>胡五斌</td></tr> </table>		Sname	1	李一在	2	刘三晶	3	杨四敏	4	胡五斌
	Sname										
1	李一在										
2	刘三晶										
3	杨四敏										
4	胡五斌										

9 查询没有选修“面向对象程序设计”课程的学生姓名 (用 NOT EXISTS)。

```
5
6 - (9) 查询没有选修“面向对象程序设计”课程的学生姓名 (用NOT EXISTS)
7 ✓ select s.Sname
8     from student s where not exists( select * from sc b where s.Sno=b.Sno and not exists(
9         select * from teaching a where a.CID=b.Cid and not exists(
10             select * from course co where co.Cno = a.Cno and co.Cname='面向对象程序设计'
11         )
12     );
```

Sname
1 李一在
2 葛畅
3 杨四敏
4 胡五斌
5 赵陈毅泽
6 王威
7 赵六玮
8 王娜
9 秦琪键
10 赵八砚
11 杨青
12 杨久玲

10 查询主讲“数据库系统”和主讲“数据结构”的教师姓名 (用 UNION)

```
(10) 查询主讲“数据库系统”和主讲“数据结构”的教师姓名 (用UNION)
SELECT Tname
FROM teacher
WHERE Tno IN (SELECT Tno FROM teaching WHERE Cno = (select cno from course where Cname='数据库系统' ))
UNION
SELECT Tname
FROM teacher
WHERE Tno IN (SELECT Tno FROM teaching WHERE Cno = (select cno from course where Cname='数据结构'));
```

Tname
张朋
徐军
王山

11 查询讲授了所有课程的教师的姓名

对数据进行了插入操作, 让查询结果有内容

WHERE					ORDER BY				
	CID	Tno	Cno	Language					
1	1	000001	800001	英文					
2	2	000002	800002	中文					
3	3	000002	800003	双语					
4	4	000003	810011	中文					
5	5	000004	810013	英文					
6	6	000005	800001	中文					
7	7	000006	800002	中文					
8	8	000007	800003	英文					
9	9	000007	810011	英文					
10	10	000008	810013	双语					
11	11	000002	800001	英文					
12	12	000002	810011	中文					
13	13	000002	810013	双语					
14	14	000002	810015	中文					

```
-- (11) 查询讲授了所有课程的教师的姓名

-- 该结果为空，所以我选择自行插入数据让数据满足要求

SELECT Tname
FROM teacher
WHERE Tno IN (
    SELECT Tno
    FROM teaching
    GROUP BY Tno
    HAVING COUNT(DISTINCT Cno) = (SELECT COUNT(DISTINCT Cno) FROM course)
);
```

Tname
1 王山

12 查询同时选修学课程 800001 和 800002 的女同学的姓名

The screenshot displays a database query tool interface. The top section shows a table with 19 rows, filtered by a WHERE clause and ordered by Cid. The table columns are Sno, Cid, and Grade. The bottom section shows a SQL query and its result.

	Sno	Cid	Grade
1	22309010	1	69
2	22301056	1	80
3	22309100	1	80
4	22309100	2	82
5	22309015	2	79
6	22301056	2	80
7	22309012	3	88
8	22309015	3	<null>
9	22309015	3	<null>
10	22301056	3	80
11	22309030	3	78
12	22309020	3	95
13	22309100	4	67
14	22309148	4	58
15	22309020	4	88
16	22309048	8	97
17	22309075	8	91
18	22309150	9	89
19	22309160	9	71

SQL Query:

```
(12) 查询同时选修学课程800001和800002的女同学的姓名
select distinct s.Sname
  from student s
    join sc sb on sb.Sno=s.Sno join teaching t on t.CID=sb.Cid and t.Cno='800001'
    join sc sb1 on sb1.Sno=s.Sno join teaching t1 on t.CID=sb1.Cid and t1.Cno='800002'
 where s.Ssex='女';
```

Result:

Sname
田仪

13 查询有一门课程成绩为 95 分的女同学的姓名

```
-- (13) 查询有一门课程成绩为95分的女同学的姓名
select s.Sname
  from student s
 join sc sb on sb.Sno=s.Sno and sb.Grade=95
 where s.Ssex='女';
```

Sname
杨四敏

14 查询选课数量大于等于 3 门的女同学的姓名

删除了关于刘三晶的重复数据，并为其添加了数据，使其满足查询要求

WHERE				ORDER BY Cid			
	Sno	Cid	Grade				
1	22309010	1	69				
2	22301056	1	80				
3	22309100	2	82				
4	22309015	2	79				
5	22301056	2	80				
6	22309012	3	88				
7	22309015	3	<null>				
8	22309015	3	<null>				
9	22301056	3	80				
10	22309030	3	78				
11	22309020	3	95				
12	22309100	4	67				
13	22309148	4	58				
14	22309020	4	88				
15	22309015	4	80				
16	22309048	8	97				
17	22309075	8	91				
18	22309150	9	89				
19	22309160	9	71				

```

1 杨四敏
-- (14) 查询选课数量大于等于3门的女同学的姓名
-- 删除了关于刘三晶的重复数据，并为其添加了数据，使其满足要求
select s.Sname
  from student s
    join sc sb on sb.Sno=s.Sno where s.Ssex='女'
 group by s.Sname having count(sb.Cid)>=3;

```

Sname
刘三晶

15 查询平均成绩大于 80 分的男同学的姓名

```

88
89 -- (15) 查询平均成绩大于80分的男同学的姓名
90 select s.Sname
91    from student s join sc sb on sb.Sno=s.Sno
92     where s.Ssex='男'
93 group by s.Sname having avg(sb.Grade)>80;

```

Sname
葛畅
赵陈毅泽

16 查询“徐军”老师所教的每一门课程的平均成绩

```

-- (16) 查询“徐军”老师所教的每一门课程的平均成绩
select sc1.cid ,coalesce(avg(sc1.Grade),0)
  from sc sc1 where sc1.cid in (select Cid
                                from teaching
                                where Tno in
                                (select Tno
                                 from teacher
                                 where Tname = '徐军'))
 group by sc1.cid;

```

cid	coalesce(avg(sc1.Grade),0)
8	94.0000
9	80.0000

17 查询男同学每一个年龄组的人数，要求按人数升序 输出人数超过 20 人的年龄组

对数据进行了插入操作，使其都满足小于 20 岁的年龄组从而有查询结果

506	-- (17) 查询男同学每一个年龄组的人数,要求按人数升序输出人数超过20人的年龄组	WHERE	ORDER BY Sno
507	INSERT INTO student (Sno, Sname, Ssex, Sbirthday, Classno, Totalcredit, Createdby,	Sno	Sname
508	VALUES	Ssex	Sbirthday
509	('22309161', '李明', '男', '2002-08-25', 'RJ2206', 0, '22301057', 19),	Classno	Totalcredit
510	('22309162', '张三', '男', '2001-07-18', 'RJ2205', 0, '22301058', 19),		
511	('22309163', '王五', '男', '2000-06-09', 'RJ2206', 0, '22301059', 19),		
512	('22309164', '赵六', '男', '1999-05-02', 'RJ2205', 0, '22301060', 19),		
513	('22309165', '刘七', '男', '1998-04-24', 'RJ2206', 0, '22301061', 19),		
514	('22309166', '李华', '男', '1997-03-17', 'RJ2205', 0, '22301062', 19),		
515	('22309167', '王鹏', '男', '1996-02-12', 'RJ2206', 0, '22301063', 19),		
516	('22309168', '张飞', '男', '1995-01-05', 'RJ2205', 0, '22301064', 19),		
517	('22309169', '关羽', '男', '1994-12-31', 'RJ2206', 0, '22301065', 19),		
518	('22309170', '刘备', '男', '1993-11-24', 'RJ2205', 0, '22301066', 19),		
519	('22309171', '诸葛亮', '男', '1992-10-17', 'RJ2206', 0, '22301067', 19),		
520	('22309172', '孙权', '男', '1991-09-08', 'RJ2205', 0, '22301068', 19),		
521	('22309173', '周瑜', '男', '1990-08-02', 'RJ2206', 0, '22301069', 19),		
522	('22309174', '吕布', '男', '1989-07-28', 'RJ2205', 0, '22301070', 19),		
523	('22309175', '张角', '男', '1988-06-20', 'RJ2206', 0, '22301071', 19),		
524	('22309176', '貂蝉', '男', '1987-05-13', 'RJ2205', 0, '22301072', 19),		
525	('22309177', '甄姬', '男', '1986-04-05', 'RJ2206', 0, '22301073', 19),		
526	('22309178', '蔡文姬', '男', '1985-03-28', 'RJ2205', 0, '22301074', 19),		
527	('22309179', '曹操', '男', '1984-02-20', 'RJ2206', 0, '22301075', 19),		
528	('22309180', '刘备', '男', '1983-01-13', 'RJ2205', 0, '22301076', 19),		
529	('22309181', '孙权', '男', '1982-12-06', 'RJ2206', 0, '22301077', 19),		
530	('22309182', '诸葛亮', '男', '1981-11-29', 'RJ2205', 0, '22301078', 19),		
531	('22309183', '周瑜', '男', '1980-10-22', 'RJ2206', 0, '22301079', 19),		
532	('22309184', '吕布', '男', '1979-09-14', 'RJ2205', 0, '22301080', 19),		
533			

534	SELECT CASE
535	WHEN Age < 20 THEN '0-19'
536	WHEN Age >= 20 AND Age < 30 THEN '20-29'
537	WHEN Age >= 30 AND Age < 40 THEN '30-39'
538	ELSE '40+'
539	END AS AgeGroup,
540	COUNT(*) AS StudentCount
541	FROM student s WHERE s.Ssex = '男'
542	GROUP BY AgeGroup
543	HAVING COUNT(*) > 20
544	ORDER BY StudentCount ASC;
545	

AgeGroup	StudentCount
1 0-19	32

18 查询每门课程成绩都大于 90 分的学生姓名

```
545
546 -- (18) 查询每门课程成绩都大于90分的学生姓名
547 ✓ select s.Sname
548       from student s
549       join sc on sc.Sno=s.Sno
550       where not exists(
551         select * from sc
552         where sc.Sno=s.Sno and sc.Grade<90
553       ) group by s.Sname;
```

Sname

- 1 赵陈毅泽
- 2 王娜

19 查询比所有女同学年龄要大的男同学的姓名

修改了 student 中的一个女学生的年龄

WHERE										ORDER BY Sno									
	Sno	Sname	Ssex	Sbirthday	Classno	Totalcredit	Createdby	Age	Addr										
1	22301056	余旺	男	2004-04-26	RJ2202		0 22301056	19	<null>										
2	22309010	李一在	男	2005-10-01	RJ2201		0 22301056	18	<null>										
3	22309012	葛畅	男	2006-08-08	RJ2201		0 22301056	18	<null>										
4	22309015	刘三晶	女	2005-05-22	RJ2201		0 22301056	18	<null>										
5	22309020	杨四敏	女	2007-01-08	RJ2201		0 22301056	18	<null>										
6	22309030	胡五斌	男	2006-10-08	RJ2201		0 22301056	18	<null>										
7	22309048	赵陈毅泽	男	2005-06-06	RJ2202		0 22301056	18	<null>										
8	22309050	王威	男	2005-06-10	RJ2202		0 22301056	18	<null>										
9	22309067	赵六玮	女	2005-08-21	RJ2203		0 22301056	18	<null>										
10	22309075	王娜	女	2006-09-23	RJ2203		0 22301056	18	<null>										
11	22309088	蔡琪健	男	2005-03-01	RJ2203		0 22301056	18	<null>										
12	22309100	田牧	女	2006-02-26	RJ2204		0 22301056	18	<null>										
13	22309148	赵八祝	男	2005-04-25	RJ2205		0 22301056	18	<null>										
14	22309150	杨青	女	2005-11-15	RJ2205		0 22301056	18	<null>										
15	22309160	杨久玲	女	2005-12-12	RJ2206		0 22301056	18	<null>										

```
554
555 -- (19) 查询比所有女同学年龄要大的男同学的姓名
556 ✓ SELECT s.Sname,s.Age
557 FROM student s
558 WHERE s.Ssex = '男'
559 AND Age >
560 (SELECT MAX(s2.Age)
561 FROM student s2 WHERE s2.Ssex = '女');
562
```

Sname	Age
余旺	19

20 查询未选修 800002 课程的女同学的姓名

```
-- (20) 查询未选修800002课程的女同学的姓名
✓ select s.Sname
   from student s where s.Ssex='女' and not exists(
       select * from sc where sc.Sno=s.Sno and Cid in (select Cid from teaching
                                                       where Cno='800002')
   );
```

Sname
杨四敏
赵六玮
王娜
杨青
杨久玲

21 查询所有课程成绩都及格的学生姓名

```
570 -- (21) 查询所有课程成绩都及格的学生姓名
571 ✓ ~ select s.Sname
572      from student s
573      join sc on sc.Sno=s.Sno
574 group by s.Sname
575 having min(sc.Grade)>=60;
```

	Sname
1	李一在
2	田仪
3	刘三晶
4	葛畅
5	赵陈毅泽
6	王娜
7	杨四敏
8	杨青
9	杨久玲
10	余旺
11	胡五斌

22 查询选修了所有课程的学生姓名

由于课程号 810015 没有该课，所以需要先添加课程，然后对 22301056 的学生添加选修使其满足选修所有课程的条件

11 Rows | Tx: Auto | DDL

	WHERE	ORDER BY		
	CID	Tno	Cno	Language
1	1	000001	800001	英文
2	2	000002	800002	中文
3	3	000002	800003	双语
4	4	000003	810011	中文
5	5	000004	810013	英文
6	6	000005	800001	中文
7	7	000006	800002	中文
8	8	000007	800003	英文
9	9	000007	810011	英文
10	10	000008	810013	双语
11	11	000008	810015	中文

student

course

sc

teaching

teacher

class

21 rows

Tx: Auto

DDL

CSV

WHERE

ORDER BY Cid

	Sno	Cid	Grade
1	22309010	1	69
2	22301056	1	80
3	22301056	11	60
4	22309100	2	82
5	22309015	2	79
6	22301056	2	80
7	22309012	3	88
8	22309015	3	<null>
9	22309015	3	<null>
10	22301056	3	80
11	22309030	3	78
12	22309020	3	95
13	22309100	4	67
14	22309148	4	58
15	22309020	4	88
16	22301056	4	60
17	22301056	5	60
18	22309048	8	97
19	22309075	8	91
20	22309150	9	89
21	22309160	9	71

585

-- (22) 查询选修了所有课程的学生姓名

586

-- 由于课程号810015没有该课，所以需要先添加课程，然后对22301056的学生添加选修使其满足选修所有课程的条件

587

select s.Sname

588

from student s

589

where Sno in (

590

select sno from sc

591

join teaching t on t.cid=sc.Cid

592

group by sno

593

having count(distinct Cno)=(select count(distinct Cno)from course)

594

);

595

> SELECT Tname FROM teacherteacher...;

603

Sname

1 余旺

23 查询选修了“葛畅”同学所选修的所有课程的学生姓名

这里默认了每个学生都只能选择一次 cid，因为陈老师已经让删去了 sc 中的重复数据

```
604 -- (23) 查询选修了“葛畅”同学所选修的所有课程的学生姓名
605 -- 这里默认了每个学生都只能选择一次cid，之前老师让删去了sc中的重复数据
606 ✓ SELECT DISTINCT s.Sname
607 FROM student s
608 JOIN sc s1 ON s.Sno = s1.Sno
609 WHERE s1.Cid IN (
610     SELECT Cid FROM sc
611     WHERE Sno = (
612         SELECT Sno FROM student WHERE Sname = '葛畅'
613     )
614 ) GROUP BY s.Sno, s.Sname
615 HAVING COUNT(DISTINCT s1.Cid) = (
616     SELECT COUNT(DISTINCT Cid) FROM sc
617     WHERE Sno = (
618         SELECT Sno FROM student WHERE Sname = '葛畅'
619     )
620 );
```

Sname
1 余旺
2 刘三晶
3 杨四敏
4 胡五斌
5 葛畅

24 查询平均成绩最高的学生姓名

```
22  
23 -- (24) 查询平均成绩最高的学生姓名  
24 ✓ SELECT s.Sname  
25 FROM student s  
26 ORDER BY (  
27     SELECT AVG(Grade)  
28     FROM sc  
29     WHERE Sno = s.Sno  
30 ) DESC  
31 LIMIT 1;;
```

	Sname
1	赵陈毅泽

25 找出个人平均成绩比所在班级平均成绩高的学生信息

```
632
633 -- (25) 找出个人平均成绩比所在班级平均成绩高的学生信息
634 ✓ SELECT s.Sname, s.Sno, s.Classno
635 FROM student s
636 JOIN (
637     SELECT Classno, AVG(Grade) AS AvgGrade
638     FROM sc
639     JOIN student ON sc.Sno = student.Sno
640     GROUP BY Classno
641 ) AS ClassAvg ON s.Classno = ClassAvg.Classno
642 JOIN (
643     SELECT Sno, AVG(Grade) AS AvgGrade
644     FROM sc
645     GROUP BY Sno
646 ) AS PersonalAvg ON s.Sno = PersonalAvg.Sno
647 WHERE PersonalAvg.AvgGrade > ClassAvg.AvgGrade;
```

	Sname	Sno	Classno
1	葛畅	22309012	RJ2201
2	赵陈毅泽	22309048	RJ2202
3	杨四敏	22309020	RJ2201
4	杨青	22309150	RJ2205

使用 SQL DML 命令完成下列对 3 张表 Student、Course、SC 的各种更新操作，并查询显示测试结果

26 将选修“徐军”老师所教所有课堂的同学的成绩提高 0.05

首先对于提高后成绩大于 100 的按 100 进行赋值，其次考虑空值不进行赋值，其中 cid 为 8 和 9 的会更新

```

5  -- (1) 将选修“徐军”老师所教所有课堂的同学的成绩提高5%
6  -- 首先对于提高后成绩大于100的按100进行赋值，其次考虑空值不进行赋值，其中cid为8和9的会更新
7  ✓ UPDATE sc
8     SET Grade = CASE
9         WHEN Grade * 1.05 > 100 THEN 100
10        ELSE Grade * 1.05
11        END
12  WHERE Cid IN (
13      SELECT Cid
14      FROM teaching
15      WHERE Tno = (
16          SELECT Tno
17          FROM teacher
18          WHERE Tname = '徐军'
19      )
20  ) AND Grade IS NOT NULL;
21  ✓ select * from sc;

```

	Sno	Cid	Grade
7	22309048	8	100
8	22309075	8	96
9	22309100	4	67
10	22309148	4	58
11	22309020	4	88
12	22309150	9	93
13	22309160	9	75
14	22301056	1	80

27 在基本表 Student 中检索每一门课程成绩都大于等于 80 分的学生学号、姓名、性别，并把检索到的值送往另一个已存在的基本表 STUD (S#, SNAME, SEX)

由于无法创建 S# 的字段名，所以用 SNO 代替

console x

work console

Playground

2 4 1 39

WHERE

SNO SNAME SEX

1 22309012 葛杨 男

2 22309048 赵陈毅泽 男

3 22309075 王娜 女

4 22309020 杨四敏 女

5 22309150 杨青 女

```

687
688
689
690 -- (2) 在基本表Student中检索每一门课程成绩都大于等于80分的学生学号、姓名、性别，并把
691 -- 由于无法创建S#的字段名，所以用SNO代替
692 ✓ CREATE TABLE STUD (
693     SNO VARCHAR(8),
694     SNAME VARCHAR(8),
695     SEX CHAR(1)
696 );
697
698 ✓ INSERT INTO STUD (SNO, SNAME, SEX)
699     SELECT s.Sno, s.Sname, s.Ssex
700     FROM student s
701     JOIN sc ON s.Sno = sc.Sno
702     GROUP BY s.Sno, s.Sname, s.Ssex
703     HAVING MIN(sc.Grade) >= 80;
704
705

```

28 在基本表 SC 中删除尚无成绩的选课记录。

console x

work console

Playground

2 4 1 39

WHERE

Sno Cid Grade

1 22309010 1 69

2 22309100 2 82

3 22309015 2 79

4 22309012 3 88

5 22309048 8 100

6 22309075 8 96

7 22309100 4 67

8 22309148 4 58

9 22309020 4 88

10 22309150 9 93

11 22309160 9 75

12 22301056 1 80

13 22301056 2 80

14 22301056 3 80

15 22309030 3 78

16 22309020 3 95

17 22301056 4 60

18 22301056 5 60

19 22301056 11 60

```

69 SELECT s.Sno, s.Sname, s.Ssex
70 FROM student s
71 JOIN sc ON s.Sno = sc.Sno
72 GROUP BY s.Sno, s.Sname, s.Ssex
73 HAVING MIN(sc.Grade) >= 80;
74
75 -- (3) 在基本表SC中删除尚无成绩的选课记录。
76
77 ✓ DELETE FROM sc
78 WHERE Grade IS NULL;
79 ✓ select * from sc;
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```


29 把“王威”同学的学习选课和成绩全部删除。

由于没有王威同学的成绩，我们先添加后删除

	student	sc ×	class	course
	20 rows			
WHERE	ORDER BY			
	Sno	Cid	Grade	
1	22309010	1	69	
2	22309100	2	82	
3	22309015	2	79	
4	22309012	3	88	
5	22309048	8	100	
6	22309075	8	96	
7	22309100	4	67	
8	22309148	4	58	
9	22309020	4	88	
10	22309150	9	93	
11	22309160	9	75	
12	22301056	1	80	
13	22301056	2	80	
14	22301056	3	80	
15	22309030	3	78	
16	22309020	3	95	
17	22301056	4	60	
18	22301056	5	60	
19	22301056	11	60	
20	22309050	12	22	

```
710 -- (4) 把“王威”同学的学习选课和成绩全部删除。
711 -- 由于没有王威同学的成绩，我们先添加后删除
712 ✓ DELETE FROM sc
713 WHERE Sno = (
714     SELECT Sno
715     FROM student
716     WHERE Sname = '王威'
717 );
718 ✓ select * from sc;
```

	Sno	Cid	Grade
6	22309075	8	96
7	22309100	4	67
8	22309148	4	58
9	22309020	4	88
10	22309150	9	93
11	22309160	9	75
12	22301056	1	80
13	22301056	2	80
14	22301056	3	80
15	22309030	3	78
16	22309020	3	95
17	22301056	4	60
18	22301056	5	60
19	22301056	11	60

30 把选修数据结构课不及格的成绩全改为空值。

添加数据来检验结果

	Sno	Cid	Grade
1	22309010	1	69
2	22309100	2	82
3	22309015	2	79
4	22309012	3	88
5	22309048	8	100
6	22309075	8	96
7	22309100	4	67
8	22309148	4	58
9	22309020	4	88
10	22309150	9	93
11	22309160	9	75
12	22301056	1	80
13	22301056	2	80
14	22301056	3	80
15	22309030	3	78
16	22309020	3	95
17	22301056	4	60
18	22301056	5	60
19	22301056	11	60
20	22301056	3	58

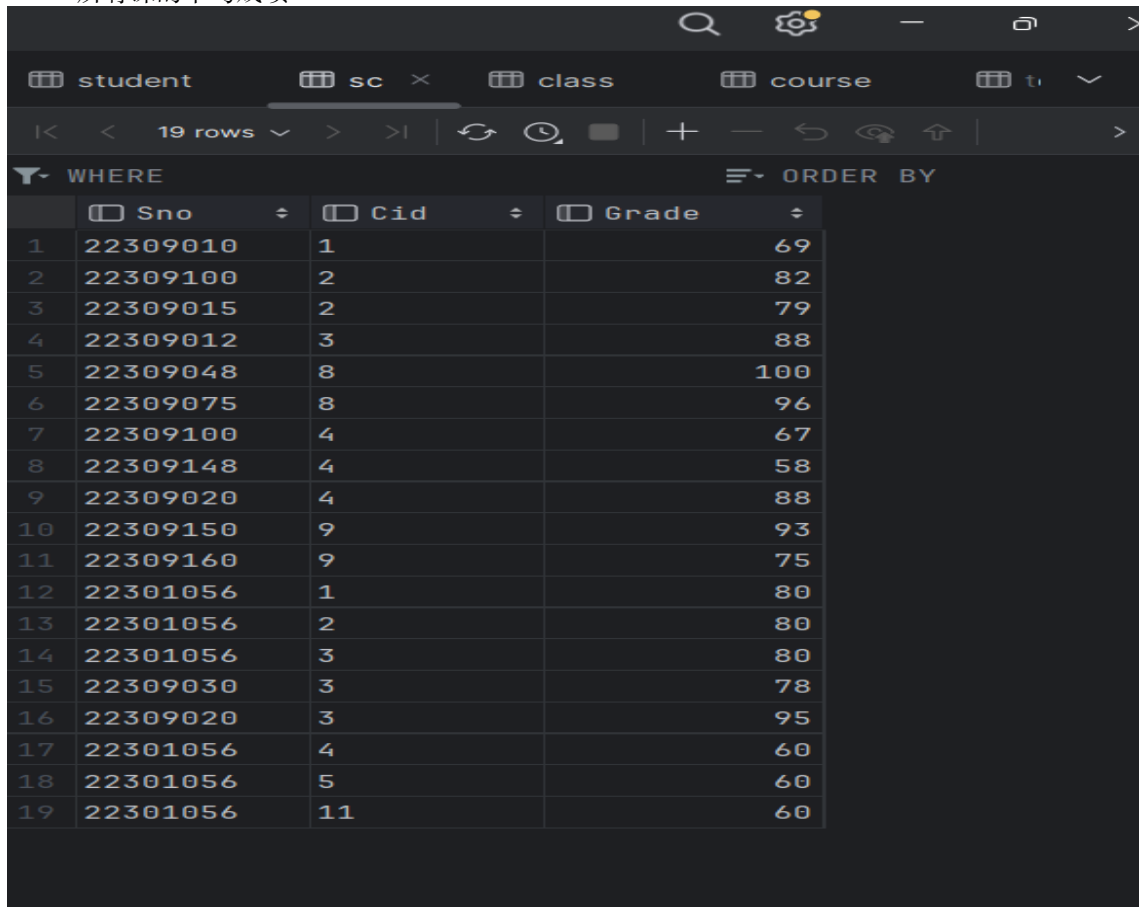
```
-- (5) 把选修数据结构课不及格的成绩全改为空值。
-- 添加数据来检验结果
UPDATE sc
SET Grade = NULL
WHERE Cid in (select teaching.Cid from teaching where teaching.Cno in
              (select course.Cno from course where Cname='数据结构'
               AND Grade < 60;
select * from sc;
```

	Sno	Cid	Grade
12	22301056	1	80
13	22301056	2	80
14	22301056	3	80
15	22309030	3	78
16	22309020	3	95
17	22301056	4	60
18	22301056	5	60
19	22301056	11	60
20	22301056	3	<null>

```
-- (6) 把低于总平均成绩的女同学的成绩提高5%
```

31 把低于总平均成绩的女同学的成绩提高 0.05

这里总平均成绩定义为所有学生所有课程的平均成绩，女同学的平均成绩为其所有课的平均成绩



	Sno	Cid	Grade
1	22309010	1	69
2	22309100	2	82
3	22309015	2	79
4	22309012	3	88
5	22309048	8	100
6	22309075	8	96
7	22309100	4	67
8	22309148	4	58
9	22309020	4	88
10	22309150	9	93
11	22309160	9	75
12	22301056	1	80
13	22301056	2	80
14	22301056	3	80
15	22309030	3	78
16	22309020	3	95
17	22301056	4	60
18	22301056	5	60
19	22301056	11	60

```
757 -- 创建临时表存储符合条件的学生学号
758 CREATE TEMPORARY TABLE IF NOT EXISTS temp_students (
759   Sno VARCHAR(8)
760 );
761 -- 计算总平均成绩
762 SET @overall_avg := (
763   SELECT AVG(Grade) FROM sc
764 );
765 -- 将低于总平均成绩的女同学的学号存储到临时表中
766 INSERT INTO temp_students (Sno)
767 SELECT student.Sno
768 FROM student
769 JOIN sc ON student.Sno = sc.Sno
770 WHERE student.Ssex = '女'
771 GROUP BY student.Sno
772 HAVING AVG(sc.Grade) < @overall_avg;
773 -- 更新符合条件的学生的成绩
774 UPDATE sc
775 SET Grade = LEAST(Grade * 1.05, 100)
776 WHERE Sno IN (SELECT Sno FROM temp_students);
777 -- 删除临时表
778 DROP TEMPORARY TABLE IF EXISTS temp_students;
```

Sno	Cid	Grade
22309010	1	69
22309100	2	86
22309015	2	79
22309012	3	88
22309048	8	100
22309075	8	96
22309100	4	70
22309148	4	58
22309020	4	88
22309150	9	93
22309160	9	79
22301056	1	80
22301056	2	80
22301056	3	80
22309030	3	78
22309020	3	95
22301056	4	60
22301056	5	60
22301056	11	60

32 在基本表 SC 中修改课程号为 800004 的课程成绩，若成绩小于等于 75 分时提高 5%，若成绩大于 75 分时提高 4% 但不能超过 100 分（用两个 UPDATE 语句实现）。成绩在基表中该字段类型为 Integer，会自动取整。

由于初始的数据无法展示更新效果，我们手动添加了数据

添加 800004 为算法设计与分析，其 CID 为 12

为不影响结果，先更新大于 75 的分数，这样就不会造成先更新小于 75 却使其分数大于 75 而执行了大于 75 分语句

Console

Tx: Auto ✓ Playground

work console

```
786 -- 成绩在基表中该字段类型为 Integer, 会自动取整。
787
788 -- 由于初始的数据无法展示更新效果, 我们手动添加了数据
789 -- 添加800004为算法设计与分析, 其CID为12
790 -- 为不影响结果, 先更新大于75的分数, 这样就不会造成先更新小于75却使其分数大于75而执行了大于75分语句
791 UPDATE sc
792 SET Grade = LEAST(Grade * 1.04, 100)
793 WHERE Cid = (select Cid from teaching where Cno='800004') AND Grade > 75;
794
795 -- 更新成绩小于等于75分的记录
796 UPDATE sc
797 SET Grade = LEAST(Grade * 1.05, 100)
798 WHERE Cid = (select Cid from teaching where Cno='800004') AND Grade <= 75;
799
800 select * from sc;
```

	Sno	Cid	Grade
14	22301056	3	80
15	22309030	3	78
16	22309020	3	95
17	22301056	4	60
18	22301056	5	60
19	22301056	11	60
20	22301056	12	63
21	22301055	12	88