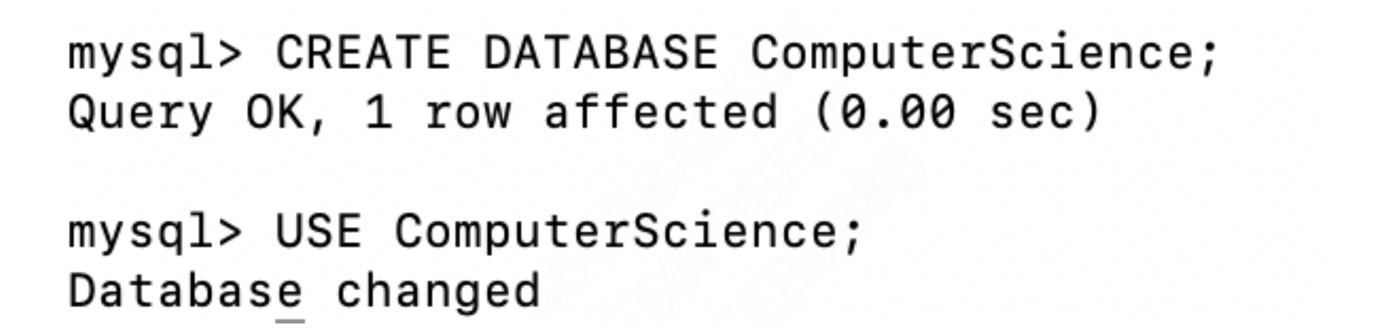
Database hw4

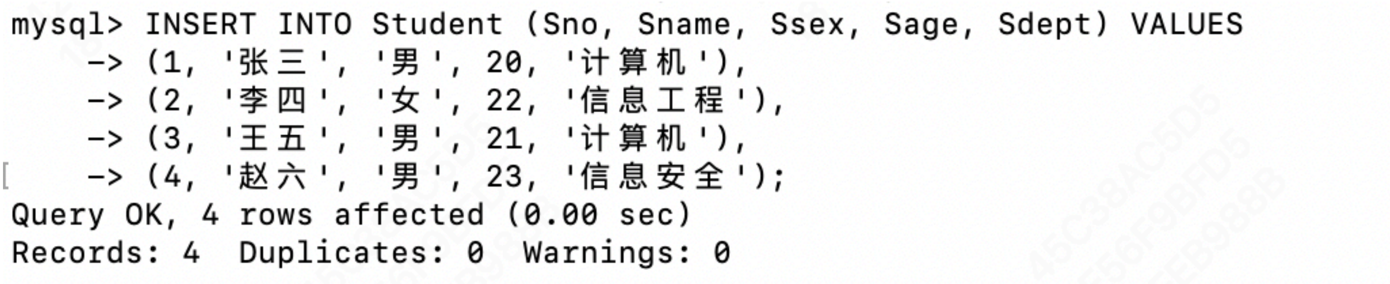
创建数据库并导入数据

```
CREATE DATABASE ComputerScience;
USE ComputerScience;
```



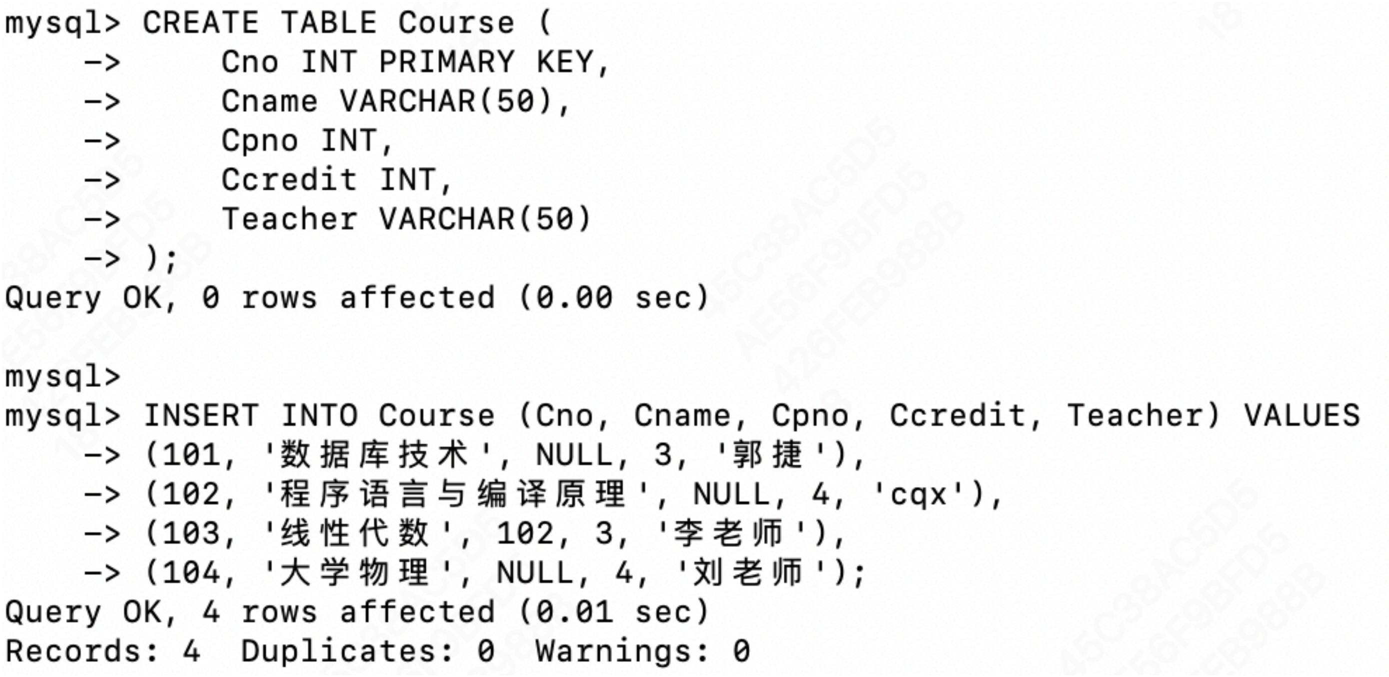
```
mysql> CREATE DATABASE ComputerScience;
Query OK, 1 row affected (0.00 sec)

mysql> USE ComputerScience;
Database changed
```

```
CREATE TABLE Student (
    Sno INT PRIMARY KEY,
    Sname VARCHAR(10),
    Ssex VARCHAR(5),
    Sage INT,
    Sdept VARCHAR(50)
);

INSERT INTO Student (Sno, Sname, Ssex, Sage, Sdept) VALUES
(1, '张三', '男', 20, '计算机'),
(2, '李四', '女', 22, '信息工程'),
(3, '王五', '男', 21, '计算机'),
(4, '赵六', '男', 23, '信息安全');
```



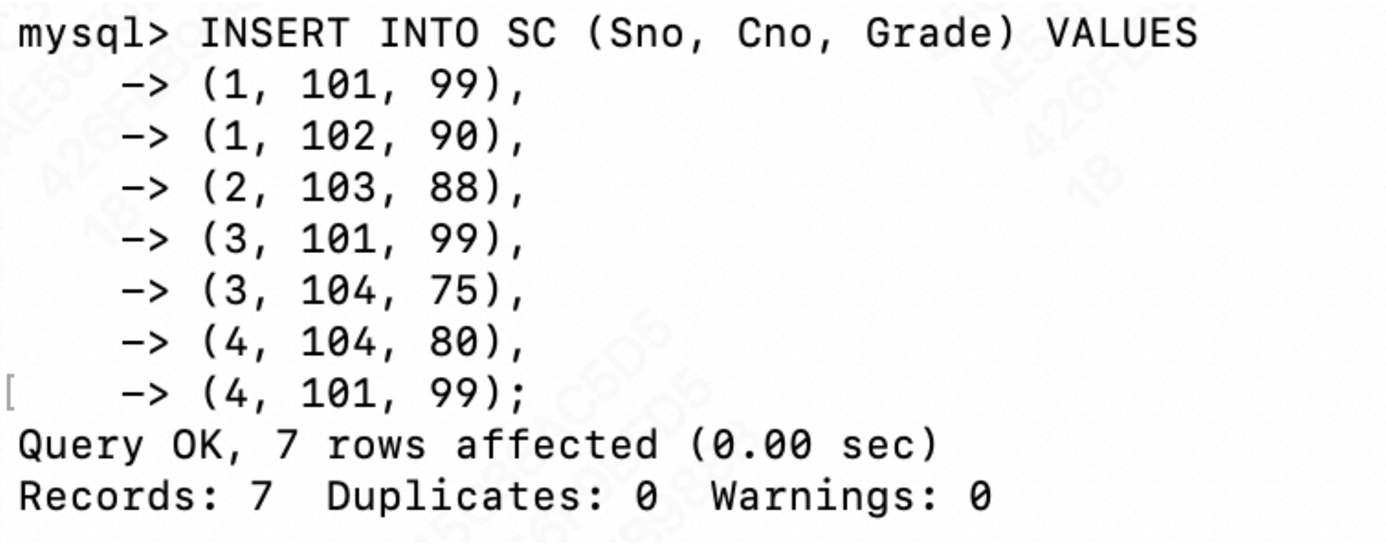
```
mysql> INSERT INTO Student (Sno, Sname, Ssex, Sage, Sdept) VALUES
    -> (1, '张三', '男', 20, '计算机'),
    -> (2, '李四', '女', 22, '信息工程'),
    -> (3, '王五', '男', 21, '计算机'),
    -> (4, '赵六', '男', 23, '信息安全');
Query OK, 4 rows affected (0.00 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

```
CREATE TABLE Course (
    Cno INT PRIMARY KEY,
    Cname VARCHAR(50),
    Cpno INT,
    Ccredit INT,
    Teacher VARCHAR(50)
);

INSERT INTO Course (Cno, Cname, Cpno, Ccredit, Teacher) VALUES
(101, '数据库技术', NULL, 3, '郭捷'),
(102, '程序语言与编译原理', NULL, 4, 'cqx'),
(103, '线性代数', 102, 3, '李老师'),
(104, '大学物理', NULL, 4, '刘老师');
```



```
mysql> CREATE TABLE Course (
    ->     Cno INT PRIMARY KEY,
    ->     Cname VARCHAR(50),
    ->     Cpno INT,
    ->     Ccredit INT,
    ->     Teacher VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> INSERT INTO Course (Cno, Cname, Cpno, Ccredit, Teacher) VALUES
    -> (101, '数据库技术', NULL, 3, '郭捷'),
    -> (102, '程序语言与编译原理', NULL, 4, 'cqx'),
    -> (103, '线性代数', 102, 3, '李老师'),
    -> (104, '大学物理', NULL, 4, '刘老师');
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

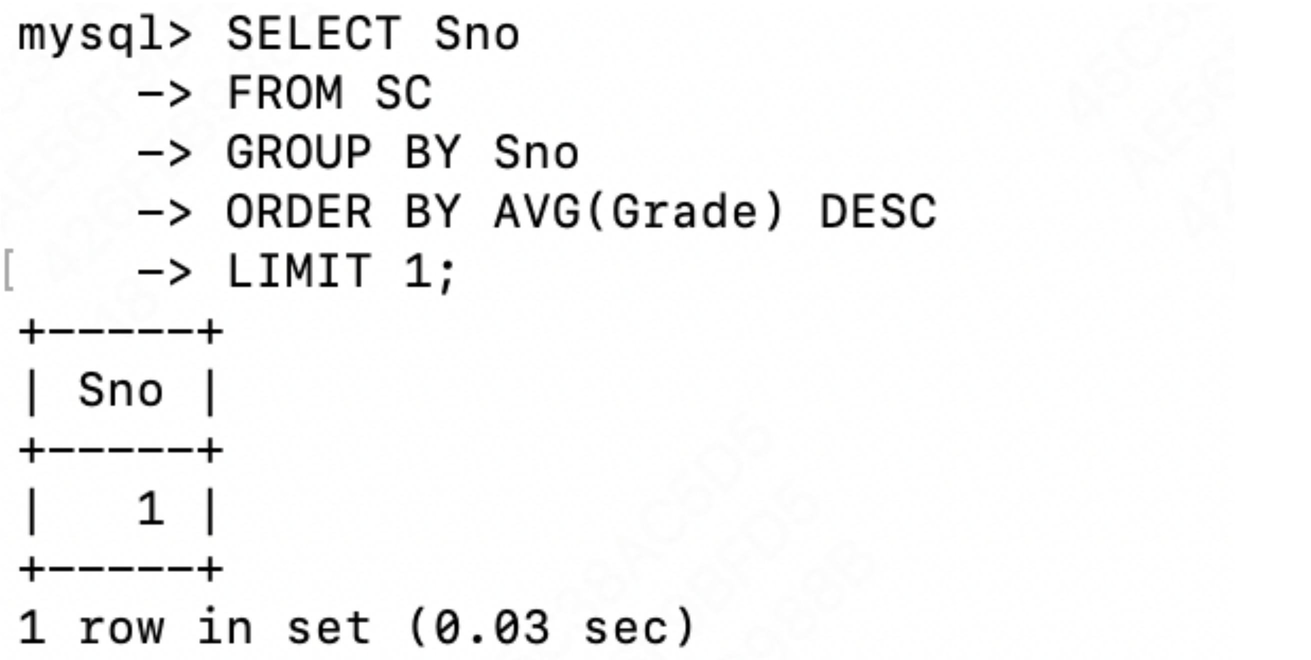
```
CREATE TABLE SC (
    Sno INT,
    Cno INT,
    Grade INT,
    PRIMARY KEY (Sno, Cno),
    FOREIGN KEY (Sno) REFERENCES Student(Sno),
    FOREIGN KEY (Cno) REFERENCES Course(Cno)
);

INSERT INTO SC (Sno, Cno, Grade) VALUES
(1, 101, 99),
(1, 102, 90),
(2, 103, 88),
(3, 101, 99),
(3, 104, 75),
(4, 104, 80),
(4, 101, 99);
```



```
mysql> INSERT INTO SC (Sno, Cno, Grade) VALUES
    -> (1, 101, 99),
    -> (1, 102, 90),
    -> (2, 103, 88),
    -> (3, 101, 99),
    -> (3, 104, 75),
    -> (4, 104, 80),
[   -> (4, 101, 99);
Query OK, 7 rows affected (0.00 sec)
Records: 7  Duplicates: 0  Warnings: 0
```

找出平均成绩最高的学生的学号

```
SELECT Sno
FROM SC
GROUP BY Sno
ORDER BY AVG(Grade) DESC
LIMIT 1;
```



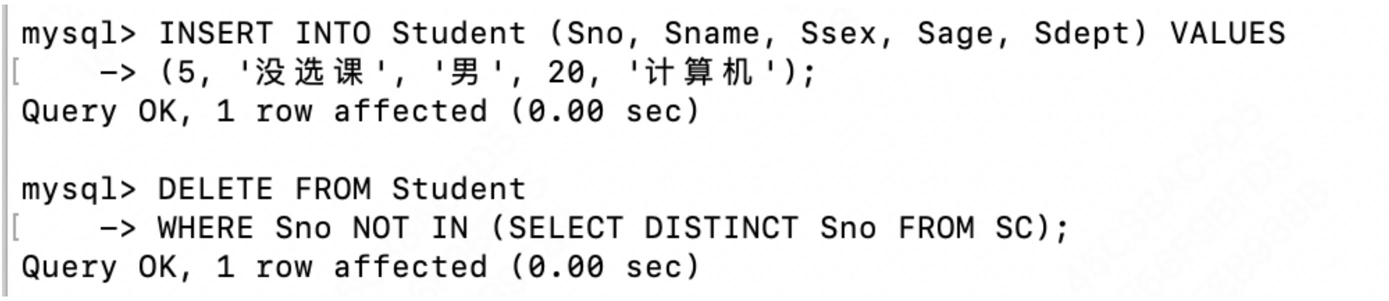
```
mysql> SELECT Sno
    -> FROM SC
    -> GROUP BY Sno
    -> ORDER BY AVG(Grade) DESC
[   -> LIMIT 1;
+-----+
| Sno |
+-----+
|   1 |
+-----+
1 row in set (0.03 sec)
```

可以看到平均分最高的学生确实为1

将没有选课的学生从学生表中删除

```
INSERT INTO Student (Sno, Sname, Ssex, Sage, Sdept) VALUES
(5, '没选课', '男', 20, '计算机');
```

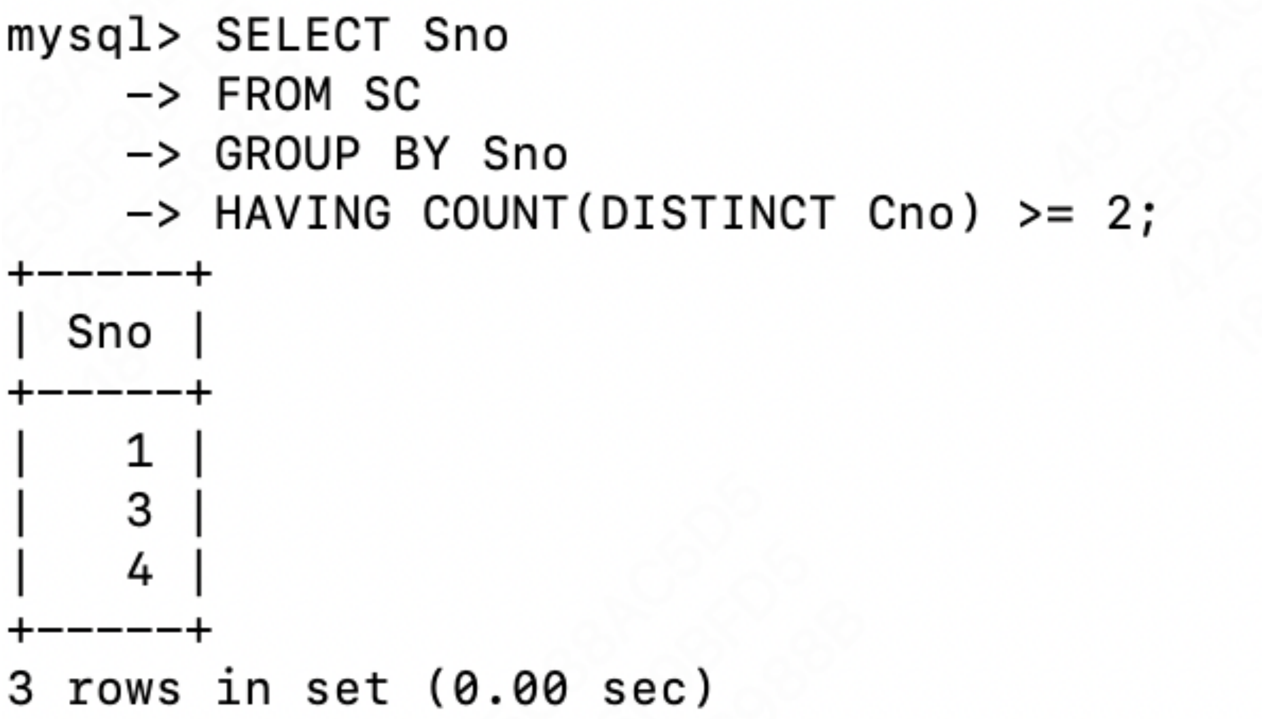
```
DELETE FROM Student
WHERE Sno NOT IN (SELECT DISTINCT Sno FROM SC);
```



```
mysql> INSERT INTO Student (Sno, Sname, Ssex, Sage, Sdept) VALUES
    -> (5, '没选课', '男', 20, '计算机');
Query OK, 1 row affected (0.00 sec)

mysql> DELETE FROM Student
    -> WHERE Sno NOT IN (SELECT DISTINCT Sno FROM SC);
Query OK, 1 row affected (0.00 sec)
```

查询出选修至少两门课程的学生学号

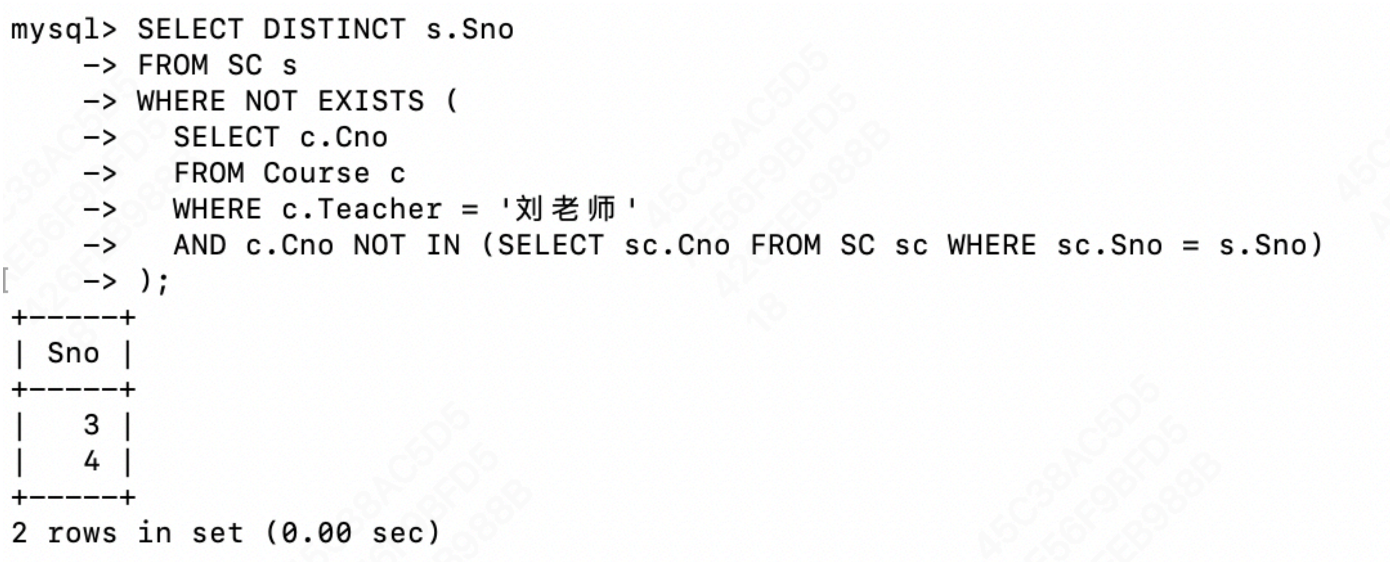
```
SELECT Sno
FROM SC
GROUP BY Sno
HAVING COUNT(DISTINCT Cno) >= 2;
```



```
mysql> SELECT Sno
    -> FROM SC
    -> GROUP BY Sno
    -> HAVING COUNT(DISTINCT Cno) >= 2;
+-----+
| Sno |
+-----+
|   1 |
|   3 |
|   4 |
+-----+
3 rows in set (0.00 sec)
```

查询选择了刘老师所有课程的学生学号

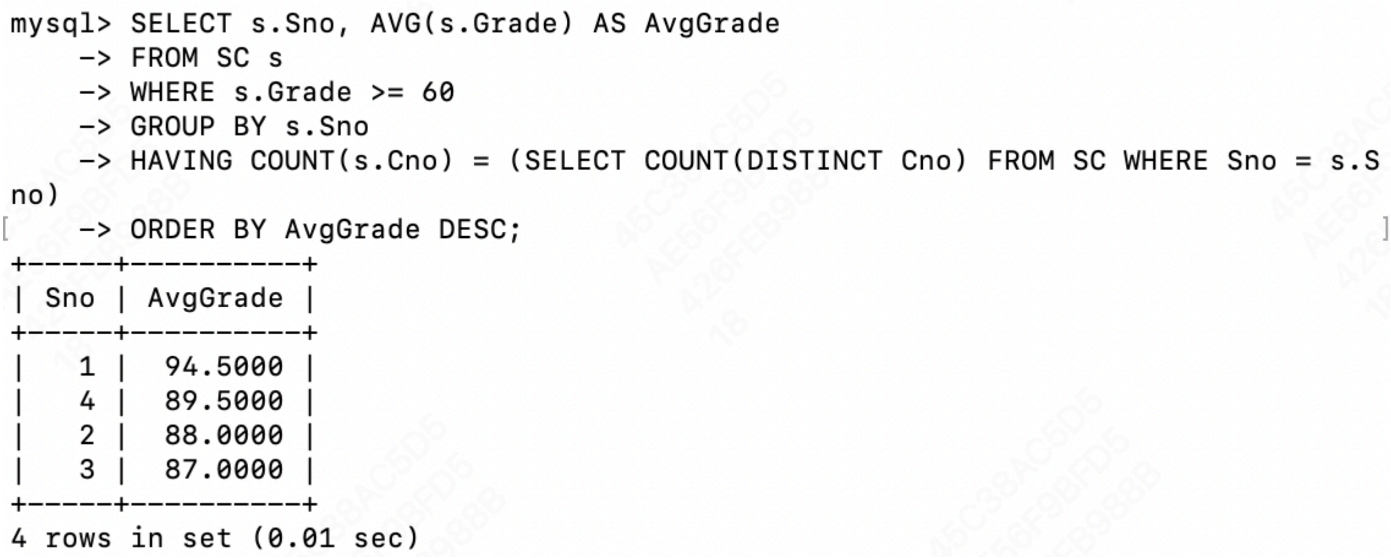
```
SELECT DISTINCT s.Sno
FROM SC s
WHERE NOT EXISTS (
  SELECT c.Cno
  FROM Course c
  WHERE c.Teacher = '刘老师'
  AND c.Cno NOT IN (SELECT sc.Cno FROM SC sc WHERE sc.Sno = s.Sno)
);
```



```
mysql> SELECT DISTINCT s.Sno
    -> FROM SC s
    -> WHERE NOT EXISTS (
    ->   SELECT c.Cno
    ->   FROM Course c
    ->   WHERE c.Teacher = '刘老师'
    ->   AND c.Cno NOT IN (SELECT sc.Cno FROM SC sc WHERE sc.Sno = s.Sno)
    -> );
+------+
| Sno  |
+------+
|    3 |
|    4 |
+------+
2 rows in set (0.00 sec)
```

按平均成绩的降序给出所有课程都及格的学生及其平均成绩

```
SELECT s.Sno, AVG(s.Grade) AS AvgGrade
FROM SC s
WHERE s.Grade >= 60
GROUP BY s.Sno
HAVING COUNT(s.Cno) = (SELECT COUNT(DISTINCT Cno) FROM SC WHERE Sno = s.Sno)
ORDER BY AvgGrade DESC;
```



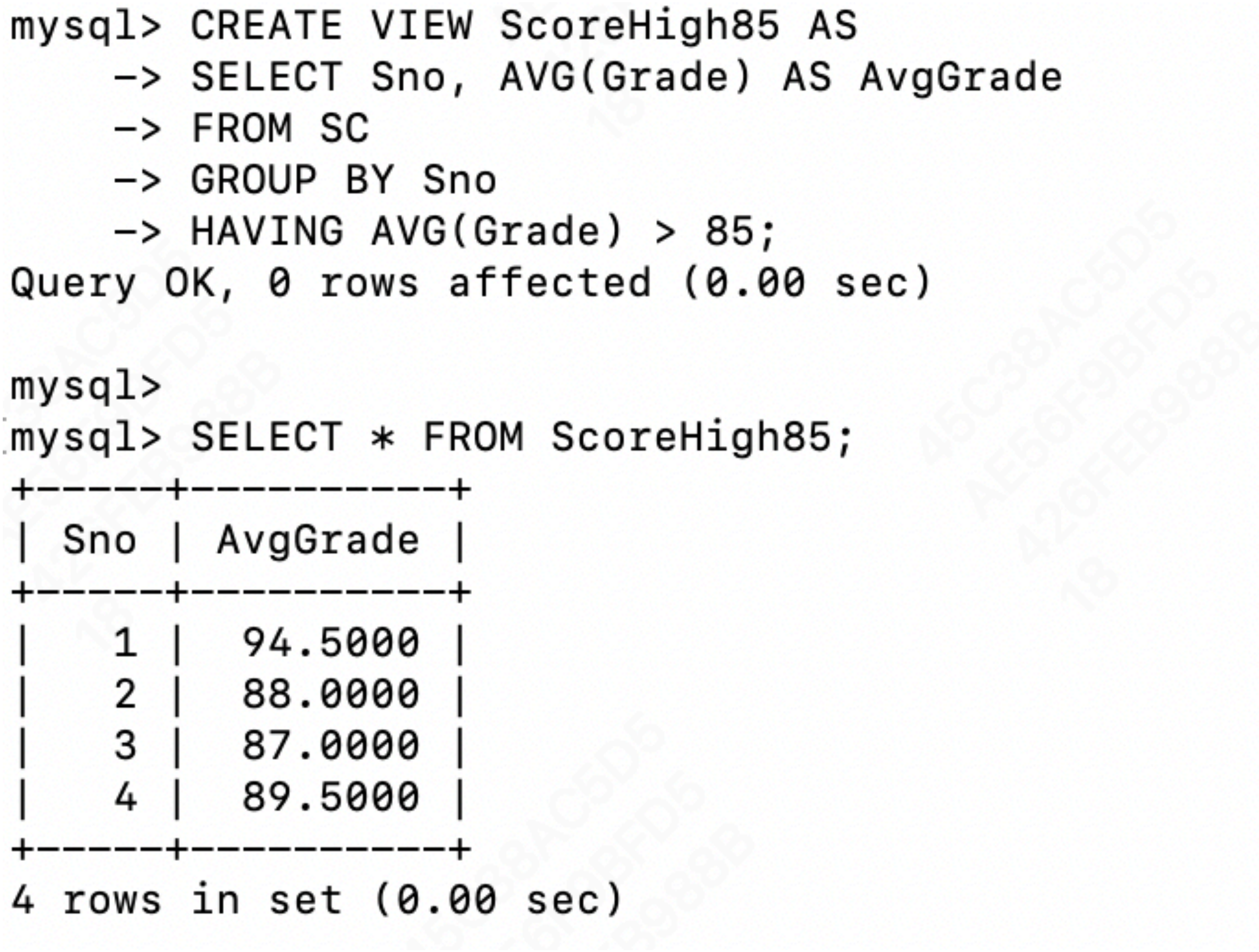
```
mysql> SELECT s.Sno, AVG(s.Grade) AS AvgGrade
    -> FROM SC s
    -> WHERE s.Grade >= 60
    -> GROUP BY s.Sno
    -> HAVING COUNT(s.Cno) = (SELECT COUNT(DISTINCT Cno) FROM SC WHERE Sno = s.S
no)
    -> ORDER BY AvgGrade DESC;
+------+----------+
| Sno  | AvgGrade |
+------+----------+
|    1 |  94.5000 |
|    4 |  89.5000 |
|    2 |  88.0000 |
|    3 |  87.0000 |
+------+----------+
4 rows in set (0.01 sec)
```

定义一个平均成绩大于85分的学生成绩视图（学号和平均成绩）

首先插入一个平均分较低的学生作为参考

```
INSERT INTO Student (Sno, Sname, Ssex, Sage, Sdept) VALUES
(5, '小明', '男', 20, '计算机');

INSERT INTO SC (Sno, Cno, Grade) VALUES
(5, 102, 81);
```

```
CREATE VIEW ScoreHigh85 AS
SELECT Sno, AVG(Grade) AS AvgGrade
FROM SC
GROUP BY Sno
HAVING AVG(Grade) > 85;

SELECT * FROM ScoreHigh85;
```



```
mysql> CREATE VIEW ScoreHigh85 AS
    -> SELECT Sno, AVG(Grade) AS AvgGrade
    -> FROM SC
    -> GROUP BY Sno
    -> HAVING AVG(Grade) > 85;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> SELECT * FROM ScoreHigh85;
+-----+----------+
| Sno | AvgGrade |
+-----+----------+
|   1 |  94.5000 |
|   2 |  88.0000 |
|   3 |  87.0000 |
|   4 |  89.5000 |
+-----+----------+
4 rows in set (0.00 sec)
```

视图中没有显示小明，查询成功

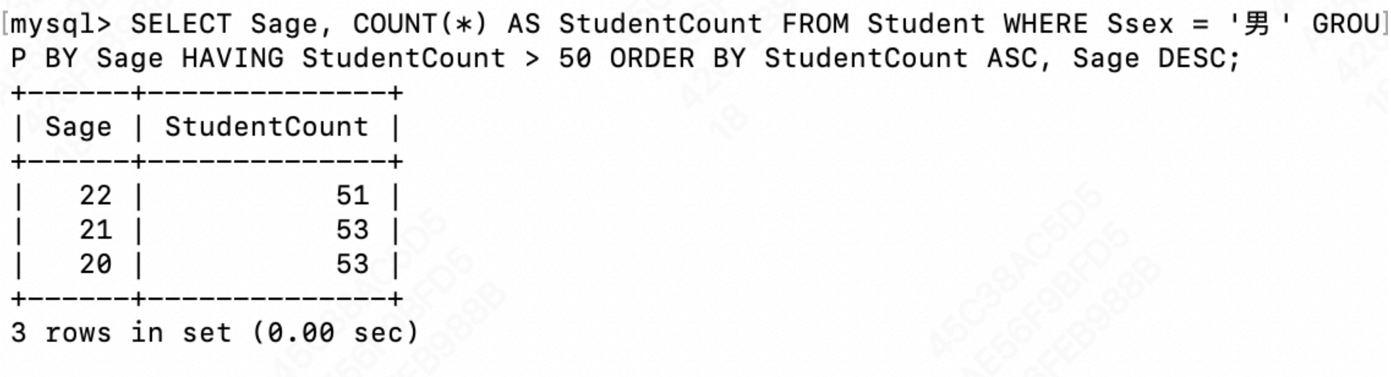
表S中男同学的每一年龄组（超过50人）有多少人？要求查询结果按人数升序排列，人数相同按年龄降序排列

首先给各个年龄组插入50+人

```
INSERT INTO Student (Sno, Sname, Ssex, Sage, Sdept)
VALUES
(6, '学生A', '男', 20, '数学'),
(7, '学生B', '男', 20, '数学'),
(8, '学生C', '男', 20, '数学'),
(9, '学生D', '男', 20, '数学'),
(10, '学生E', '男', 20, '数学'),
(11, '学生F', '男', 20, '数学'),
(12, '学生G', '男', 20, '数学'),
(13, '学生H', '男', 20, '数学'),
(14, '学生I', '男', 20, '数学'),
(15, '学生J', '男', 20, '数学'),
(16, '学生K', '男', 20, '数学'),
(17, '学生L', '男', 20, '数学'),
(18, '学生M', '男', 20, '数学'),
(19, '学生N', '男', 20, '数学'),
(20, '学生O', '男', 20, '数学'),
(21, '学生P', '男', 20, '数学'),
(22, '学生Q', '男', 20, '数学'),
(23, '学生R', '男', 20, '数学'),
(24, '学生S', '男', 20, '数学'),
(25, '学生T', '男', 20, '数学'),
(26, '学生U', '男', 20, '数学'),
(27, '学生V', '男', 20, '数学'),
(28, '学生W', '男', 20, '数学'),
(29, '学生X', '男', 20, '数学'),
(30, '学生Y', '男', 20, '数学'),
(31, '学生Z', '男', 20, '数学'),
(32, '学生AA', '男', 20, '数学'),
(33, '学生BB', '男', 20, '数学'),
(34, '学生CC', '男', 20, '数学'),
(35, '学生DD', '男', 20, '数学'),
(36, '学生EE', '男', 20, '数学'),
(37, '学生FF', '男', 20, '数学'),
(38, '学生GG', '男', 20, '数学'),
(39, '学生HH', '男', 20, '数学'),
(40, '学生II', '男', 20, '数学'),
(41, '学生JJ', '男', 20, '数学'),
(42, '学生KK', '男', 20, '数学'),
(43, '学生LL', '男', 20, '数学'),
(44, '学生MM', '男', 20, '数学'),
(45, '学生NN', '男', 20, '数学'),
(46, '学生OO', '男', 20, '数学'),
(47, '学生PP', '男', 20, '数学'),
(48, '学生QQ', '男', 20, '数学'),
(49, '学生RR', '男', 20, '数学'),
(50, '学生SS', '男', 20, '数学'),
(51, '学生TT', '男', 20, '数学'),
(52, '学生UU', '男', 20, '数学'),
(53, '学生VV', '男', 20, '数学'),
(54, '学生WW', '男', 20, '数学'),
(55, '学生XX', '男', 20, '数学'),
(56, '学生YY', '男', 20, '数学');


INSERT INTO Student (Sno, Sname, Ssex, Sage, Sdept)
VALUES
(57, '学生A', '男', 21, '数学'),
(58, '学生B', '男', 21, '数学'),
(59, '学生C', '男', 21, '数学'),
(60, '学生D', '男', 21, '数学'),
(61, '学生E', '男', 21, '数学'),
(62, '学生F', '男', 21, '数学'),
(63, '学生G', '男', 21, '数学'),
(64, '学生H', '男', 21, '数学'),
(65, '学生I', '男', 21, '数学'),
(66, '学生J', '男', 21, '数学'),
(67, '学生K', '男', 21, '数学'),
(68, '学生L', '男', 21, '数学'),
(69, '学生M', '男', 21, '数学'),
(70, '学生N', '男', 21, '数学'),
(71, '学生O', '男', 21, '数学'),
(72, '学生P', '男', 21, '数学'),
(73, '学生Q', '男', 21, '数学'),
(74, '学生R', '男', 21, '数学'),
(75, '学生S', '男', 21, '数学'),
(76, '学生T', '男', 21, '数学'),
(77, '学生U', '男', 21, '数学'),
(78, '学生V', '男', 21, '数学'),
(79, '学生W', '男', 21, '数学'),
(80, '学生X', '男', 21, '数学'),
(81, '学生Y', '男', 21, '数学'),
(82, '学生Z', '男', 21, '数学'),
(83, '学生AA', '男', 21, '数学'),
(84, '学生BB', '男', 21, '数学'),
(85, '学生CC', '男', 21, '数学'),
(86, '学生DD', '男', 21, '数学'),
(87, '学生EE', '男', 21, '数学'),
(88, '学生FF', '男', 21, '数学'),
(89, '学生GG', '男', 21, '数学'),
(90, '学生HH', '男', 21, '数学'),
(91, '学生II', '男', 21, '数学'),
(92, '学生JJ', '男', 21, '数学'),
(93, '学生KK', '男', 21, '数学'),
(94, '学生LL', '男', 21, '数学'),
(95, '学生MM', '男', 21, '数学'),
(96, '学生NN', '男', 21, '数学'),
(97, '学生OO', '男', 21, '数学'),
(98, '学生PP', '男', 21, '数学'),
(99, '学生QQ', '男', 21, '数学'),
(100, '学生RR', '男', 21, '数学'),
(101, '学生SS', '男', 21, '数学'),
(102, '学生TT', '男', 21, '数学'),
(103, '学生UU', '男', 21, '数学'),
(104, '学生VV', '男', 21, '数学'),
(105, '学生WW', '男', 21, '数学'),
(106, '学生XX', '男', 21, '数学'),
(159, '学生XX', '男', 21, '数学'),
(107, '学生YY', '男', 21, '数学');
```

```sql
INSERT INTO Student (Sno, Sname, Ssex, Sage, Sdept)
VALUES
(108, '学生A', '男', 22, '物理'),
(109, '学生B', '男', 22, '物理'),
(110, '学生C', '男', 22, '物理'),
(111, '学生D', '男', 22, '物理'),
(112, '学生E', '男', 22, '物理'),
(113, '学生F', '男', 22, '物理'),
(114, '学生G', '男', 22, '物理'),
(115, '学生H', '男', 22, '物理'),
(116, '学生I', '男', 22, '物理'),
(117, '学生J', '男', 22, '物理'),
(118, '学生K', '男', 22, '物理'),
(119, '学生L', '男', 22, '物理'),
(120, '学生M', '男', 22, '物理'),
(121, '学生N', '男', 22, '物理'),
(122, '学生O', '男', 22, '物理'),
(123, '学生P', '男', 22, '物理'),
(124, '学生Q', '男', 22, '物理'),
(125, '学生R', '男', 22, '物理'),
(126, '学生S', '男', 22, '物理'),
(127, '学生T', '男', 22, '物理'),
(128, '学生U', '男', 22, '物理'),
(129, '学生V', '男', 22, '物理'),
(130, '学生W', '男', 22, '物理'),
(131, '学生X', '男', 22, '物理'),
(132, '学生Y', '男', 22, '物理'),
(133, '学生Z', '男', 22, '物理'),
(134, '学生AA', '男', 22, '物理'),
(135, '学生BB', '男', 22, '物理'),
(136, '学生CC', '男', 22, '物理'),
(137, '学生DD', '男', 22, '物理'),
(138, '学生EE', '男', 22, '物理'),
(139, '学生FF', '男', 22, '物理'),
(140, '学生GG', '男', 22, '物理'),
(141, '学生HH', '男', 22, '物理'),
(142, '学生II', '男', 22, '物理'),
(143, '学生JJ', '男', 22, '物理'),
(144, '学生KK', '男', 22, '物理'),
(145, '学生LL', '男', 22, '物理'),
(146, '学生MM', '男', 22, '物理'),
(147, '学生NN', '男', 22, '物理'),
(148, '学生OO', '男', 22, '物理'),
(149, '学生PP', '男', 22, '物理'),
(150, '学生QQ', '男', 22, '物理'),
(151, '学生RR', '男', 22, '物理'),
(152, '学生SS', '男', 22, '物理'),
(153, '学生TT', '男', 22, '物理'),
(154, '学生UU', '男', 22, '物理'),
(155, '学生VV', '男', 22, '物理'),
(156, '学生WW', '男', 22, '物理'),
(157, '学生XX', '男', 22, '物理'),
(158, '学生YY', '男', 22, '物理');
```

```
SELECT Sage, COUNT(*) AS StudentCount
FROM Student
WHERE Ssex = '男'
GROUP BY Sage
HAVING StudentCount > 50
ORDER BY StudentCount ASC, Sage DESC;
```



```
mysql> SELECT Sage, COUNT(*) AS StudentCount FROM Student WHERE Ssex = '男' GROU
P BY Sage HAVING StudentCount > 50 ORDER BY StudentCount ASC, Sage DESC;
+------+--------------+
| Sage | StudentCount |
+------+--------------+
|   22 |           51 |
|   21 |           53 |
|   20 |           53 |
+------+--------------+
3 rows in set (0.00 sec)
```

可以看到实现了所要求查询