知乎

50道SQL练习题

mmggqq

关注他

227 人赞同了该文章

题目转载自超经典SQL练习题，做完这些你的ＳＱＬ就过关了

下班空闲时间正好搜了一些SQL题来做，练习SQL。题目是最简单的学生选课场景。

我使用的Mysql版本是5.7.19。SQL语句可能会因数据库系统的不同会有少许差异

知乎不支持Markdown。如果觉得边幅太长可以看这个50道Mysql 练习题

数据表

--1.学生表

Student(SId,Sname,Sage,Ssex)

--SId 学生编号,Sname 学生姓名,Sage 出生年月,Ssex 学生性别

--2.课程表

Course(CId,Cname,TId)

--CId 课程编号,Cname 课程名称,TId 教师编号

--3.教师表

Teacher(TId,Tname)

--TId 教师编号,Tname 教师姓名

▲ 赞同 227 ▼ 48 条评论 分享 收

--4.成绩表

--SId 学生编号,CId 课程编号,score 分数

创建测试数据

学生表 Student

```
create table Student(SId varchar(10),Sname varchar(10),Sage datetime,Ssex varc
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('09' , '张三' , '2017-12-20' , '女');
insert into Student values('10' , '李四' , '2017-12-25' , '女');
insert into Student values('11' , '李四' , '2017-12-30' , '女');
insert into Student values('12' , '赵六' , '2017-01-01' , '女');
insert into Student values('13' , '孙七' , '2018-01-01' , '女');
```

科目表 Course

```
create table Course(CId varchar(10),Cname nvarchar(10),TId varchar(10))
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 Teacher

```
create table Teacher(TId varchar(10),Tname varchar(10))
insert into Teacher values('01' , '张三')
insert into Teacher values('02' , '李四')
insert into Teacher values('03' , '王五')
```

成绩表 SC

```
insert into SC values('01' , '02' , 90)
insert into SC values('01' , '03' , 99)
insert into SC values('02' , '01' , 70)
insert into SC values('02' , '02' , 60)
insert into SC values('02' , '03' , 80)
insert into SC values('03' , '01' , 80)
insert into SC values('03' , '02' , 80)
insert into SC values('03' , '03' , 80)
insert into SC values('04' , '01' , 50)
insert into SC values('04' , '02' , 30)
insert into SC values('04' , '03' , 20)
insert into SC values('05' , '01' , 76)
insert into SC values('05' , '02' , 87)
insert into SC values('06' , '01' , 31)
insert into SC values('06' , '03' , 34)
insert into SC values('07' , '02' , 89)
insert into SC values('07' , '03' , 98)
```

练习题目

1. 查询" 01 "课程比" 02 "课程成绩高的学生的信息及课程分数

1.1 查询同时存在" 01 "课程和" 02 "课程的情况

1.2 查询存在" 01 "课程但可能不存在" 02 "课程的情况(不存在时显示为 null)

1.3 查询不存在" 01 "课程但存在" 02 "课程的情况

2. 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

3. 查询在 SC 表存在成绩的学生信息

4. 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(没成绩的显示为 null)

4.1 查有成绩的学生信息

5. 查询「李」姓老师的数量

6. 查询学过「张三」老师授课的同学的信息

8. 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

9. 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

10. 查询没学过"张三"老师讲授的任一门课程的学生姓名

11. 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

12. 检索" 01 "课程分数小于 60，按分数降序排列的学生信息

13. 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

14. 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

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

15. 按各科成绩进行排序，并显示排名， Score 重复时保留名次空缺

15.1 按各科成绩进行排序，并显示排名， Score 重复时合并名次

16. 查询学生的总成绩，并进行排名，总分重复时保留名次空缺

16.1 查询学生的总成绩，并进行排名，总分重复时不保留名次空缺

17. 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]，[60-0] 及所占百分比

18. 查询各科成绩前三名的记录

19. 查询每门课程被选修的学生数

20. 查询出只选修两门课程的学生学号和姓名

21. 查询男生、女生人数

▲ 赞同 227 ▼ 48 条评论 分享 ★ 收

23. 查询同名同性学生名单，并统计同名人数

24. 查询 1990 年出生的学生名单

25. 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

26. 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

27. 查询课程名称为「数学」，且分数低于 60 的学生姓名和分数

28. 查询所有学生的课程及分数情况（存在学生没成绩，没选课的情况）

29. 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

30. 查询不及格的课程

31. 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

32. 求每门课程的学生人数

33. 成绩不重复，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

34. 成绩有重复的情况下，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

35. 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

36. 查询每门功成绩最好的前两名

37. 统计每门课程的学生选修人数（超过 5 人的课程才统计）。

38. 检索至少选修两门课程的学生学号

39. 查询选修了全部课程的学生信息

40. 查询各学生的年龄，只按年份来算

41. 按照出生日期来算，当前月日 < 出生年月的

▲ 赞同 227 ▼ 48 条评论 分享 收

43. 查询下周过生日的学生

44. 查询本月过生日的学生

45. 查询下月过生日的学生

自己写的参考答案，如果有错误或者更好的解答，有劳各位提供，谢谢。

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 *
from (select SId ,score from sc where sc.CId='01')as t1 , (select SId ,score f
where t1.SId=t2.SId
and   t1.score>t2.score
```

1.1 查询同时存在" 01 "课程和" 02 "课程的情况

```
select *
from (select SId ,score from sc where sc.CId='01')as t1 , (select SId ,score f
where t1.SId=t2.SId
```

1.2 查询存在" 01 "课程但可能不存在" 02 "课程的情况(不存在时显示为 null)

```
select *
from (select SId ,score from sc where sc.CId='01')as t1 left join  (select SId
on t1.SId=t2.SId
```

1.3 查询不存在" 01 "课程但存在" 02 "课程的情况

```
select *
from sc
where sc.SId not in (select SId from
and  sc.CId='02'
```

```
select student.*,t1.avgscore
from student inner JOIN(
select sc.SId ,AVG(sc.score)as avgscore
from sc
GROUP BY sc.SId
HAVING AVG(sc.score)>=60)as t1 on student.SId=t1.SId
```

3. 查询在 SC 表存在成绩的学生信息

```
select DISTINCT student.*
from student ,sc
where student.SId=sc.SId
```

4.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(没成绩的显示为null)

```
select student.SId,student.Sname,t1.sumscore,t1.coursecount
from student ,(
select SC.SId,sum(sc.score)as sumscore ,count(sc.CId) as coursecount
from sc
GROUP BY sc.SId) as t1
where student.SId =t1.SId
```

4.1 查有成绩的学生信息

```
select *
from student
where EXISTS(select * from sc where student.SId=sc.SId)
```

5. 查询「李」姓老师的数量

```
select count(*)
from teacher
where teacher.Tname like '李%
```

```
select student.*
from teacher  ,course  ,student,sc
where teacher.Tname='张三'
and   teacher.TId=course.TId
and   course.CId=sc.CId
and   sc.SId=student.SId
```

7. 查询没有学全所有课程的同学的信息

- 解法1

```
select student.*
from student
where student.SId not in ( select sc.SId
from sc
group by sc.SId
having count(*)=(select count(CId) from course))
```

- 解法2

```
select DISTINCT student.*
from
(select student.SId,course.CId
from student,course ) as t1 LEFT JOIN (SELECT sc.SId,sc.CId from sc)as t2 on t
where t2.SId is null
and   t1.SId=student.SId
```

利用笛卡尔积可以把什么课都没选的同学查询出来

8. 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 DISTINCT student.*
from  sc ,student
where sc.CId in (select CId from sc where sc.SId='01')
and   sc.SId=student.SId
```

```
select *
from student
where student.SId not in (
select t1.SId
from
(select student.SId,t.CId
from student ,(select sc.CId from sc where sc.SId='01') as t )as t1
left join sc on t1.SId=sc.SId and t1.CId=sc.CId
where sc.CId is null )
and student.SId !='01'
```

10.查询没学过"张三"老师讲授的任一门课程的学生姓名

```
select *
from student
where student.SId not in
(
select student.SId
from student left join sc on student.SId=sc.SId
where EXISTS
(select *
from teacher ,course
where teacher.Tname='张三'
and   teacher.TId=course.TId
and     course.CId=sc.CId))
```

11.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 student.SId,student.Sname,avg(sc.score)
from student ,sc
where student.SId=sc.SId
and   sc.score<60
GROUP BY sc.SId
HAVING count(*)>=2
```

12. 检索" 01 "课程分数小于 60，按分数降序排

```
where sc.CId ='01'
and   sc.score<60
and   student.SId=sc.SId
```

13. 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
sc.SId,sc.CId,sc.score,t1.avgscore
from  sc left join (select sc.SId,avg(sc.score) as avgscore
from sc
GROUP BY sc.SId) as t1 on sc.SId =t1.SId
ORDER BY t1.avgscore DESC
```

14. 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

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 sc.CId ,max(sc.score)as 最高分,min(sc.score)as 最低分,AVG(sc.score)as 平均
from sc
GROUP BY sc.CId
ORDER BY count(*)DESC,sc.CId asc
```

15. 按各科成绩进行排序，并显示排名， Score 重复时保留名次空缺

```
select sc.SId,sc.CId ,case when @pre_parent_code=sc.CId then @curRank:=@curRank
from (select @curRank:=0,@pre_parent_code:='') as t ,sc
ORDER by sc.CId,sc.score desc
```

15.1 按各科成绩进行排序，并显示排名， Score 重复时合并名次

```
when  @prefontscore:=sc.score then  @curRank:=1 end as rank ,sc.score,@pre_pare
from (select @curRank:=0,@pre_parent_code:='',@prefontscore :=null) as t ,sc
ORDER by sc.CId,sc.score desc
```

16. 查询学生的总成绩，并进行排名，总分重复时保留名次空缺

```
select t1.*,@currank:= @currank+1 as rank
from (select sc.SId, sum(score)
from sc
GROUP BY sc.SId
ORDER BY sum(score) desc) as t1,(select @currank:=0) as t
```

16.1 查询学生的总成绩，并进行排名，总分重复时不保留名次空缺

```
select t1.*, case when @fontscore=t1.sumscore then @currank  when @fontscore:=
from (select sc.SId, sum(score) as sumscore
from sc
GROUP BY sc.SId
ORDER BY sum(score) desc) as t1,(select @currank:=0,@fontscore:=null) as t
```

17. 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]，[60-0] 及所占百分比

```
select course.CId,course.Cname,t1.*
from course LEFT JOIN (
select sc.CId,CONCAT(sum(case when sc.score>=85 and sc.score<=100 then 1 else 
CONCAT(sum(case when sc.score>=70 and sc.score<85 then 1 else 0 end )/count(*)
CONCAT(sum(case when sc.score>=60 and sc.score<70 then 1 else 0 end )/count(*)
CONCAT(sum(case when sc.score>=0 and sc.score<60 then 1 else 0 end )/count(*)*
from sc
GROUP BY sc.CId) as t1 on course.CId=t1.CId
```

18. 查询各科成绩前三名的记录

思路：前三名转化为若大于此成绩的数量少于3即为前三名

```
where  (select count(*) from sc as a where sc.CId =a.CId and  sc.score <a.score
ORDER BY CId asc,sc.score desc
```

19. 查询每门课程被选修的学生数

```
select sc.CId,count(*)
from sc
GROUP BY sc.CId
```

20. 查询出只选修两门课程的学生学号和姓名

```
select student.SId,student.Sname
from sc,student
where student.SId=sc.SId
GROUP BY sc.SId
HAVING count(*)=2
```

21.查询男生、女生人数

```
select student.Ssex ,count(*) as 人数
from student
GROUP BY student.Ssex
```

22. 查询名字中含有「风」字的学生信息

```
select *
from student
where student.Sname like '%风%'
```

23.查询同名同性学生名单，并统计同名人数

```
select *
from student LEFT JOIN (select Sname,Ssex,COUNT(*)同名人数 from Student group by
on student.Sname =t1.Sname and studen
where t1.同名人数>1
```

```
select *
from student
where YEAR(student.Sage)=1990
```

25.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 sc.CId,AVG(sc.score)
from sc
GROUP BY sc.CId
ORDER BY AVG(sc.score) desc ,sc.CId asc
```

26.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 student.SId,student.Sname,t1.avgscore
from student INNER JOIN (select sc.SId ,AVG(sc.score) as avgscore from sc GROU
student.SId=t1.SId
```

27. 查询课程名称为「数学」，且分数低于 60 的学生姓名和分数

```
select student.Sname ,t1.score
from student INNER JOIN  (select sc.SId,sc.score
from sc,course
where sc.CId=course.CId
and   course.Cname='数学'
and   sc.score<60)as t1 on student.SId=t1.SId
```

28. 查询所有学生的课程及分数情况（存在学生没成绩，没选课的情况）

```
select student.SId,sc.CId,sc.score from Student  left join sc  on student.SId=
```

29. 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 student.Sname,course.Cname,sc.score
from student , sc  ,course
where sc.score>=70
```

30.查询存在不及格的课程

```
select DISTINCT sc.CId
from sc
where sc.score <60
```

31.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 student.SId,student.Sname
from student ,sc
where sc.CId='01'
and  student.SId=sc.SId
and  sc.score>80
```

32. 求每门课程的学生人数

```
select sc.CId,count(*) as 学生人数
from sc
GROUP BY sc.CId
```

33. 成绩不重复，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

```
select student.*,sc.score
from student ,course ,teacher ,sc
where course.CId=sc.CId
and course.TId=teacher.TId
and teacher.Tname='张三'
and student.SId =sc.SId
LIMIT 1
```

34. 成绩有重复的情况下，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

```
from course ,teacher ,sc,(select @fontage:=null,@rank:=0) as t
where course.CId=sc.CId
and course.TId=teacher.TId
and teacher.Tname='张三'
ORDER BY sc.score DESC) as t1 on student.SId=t1.SId
where t1.rank=1
```

35. 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 *
from sc as t1
where exists(select * from sc as t2 where t1.SId=t2.SId and t1.CId!=t2.CId and
```

36.查询每门功成绩最好的前两名

```
select *
from sc as t1
where (select count(*) from sc as t2 where t1.CId=t2.CId and t2.score >t1.score
ORDER BY t1.CId
```

37.统计每门课程的学生选修人数（超过 5 人的课程才统计）

```
select sc.CId as 课程编号,count(*) as 选修人数
from sc
GROUP BY sc.CId
HAVING count(*)>5
```

38.检索至少选修两门课程的学生学号

```
select DISTINCT t1.SId
from sc as t1
where (select count(* )from sc where t1.SId=sc.SId)>=3
```

39. 查询选修了全部课程的学生信息

▲ 赞同 227 ▼ 48 条评论 分享 收

```
where sc.SId=student.SId
GROUP BY sc.SId
HAVING count(*) = (select DISTINCT count(*) from course )
```

40.查询各学生的年龄，只按年份来算

```
select student.SId as 学生编号,student.Sname  as  学生姓名,TIMESTAMPDIFF(YEAR,stu
from student
```

41. 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 student.SId as 学生编号,student.Sname  as  学生姓名,TIMESTAMPDIFF(YEAR,stu
from student
```

42.查询本周过生日的学生

```
select *
from student
where YEARWEEK(student.Sage)=YEARWEEK(CURDATE())
```

43. 查询下周过生日的学生

```
select *
from student
where YEARWEEK(student.Sage)=CONCAT(YEAR(CURDATE()),week(CURDATE())+1)
```

44.查询本月过生日的学生

```
select *
from student
where EXTRACT(YEAR_MONTH FROM student.Sage)=EXTRACT(YEAR_MONTH FROM CURDATE())
```

45.查询下月过生日的学生