

目录

- [考点](#)
- [表定义及表记录如下](#)
- [题目](#)

考点

1. 查询：分为单表查询（123）和多表查询（456）
 1. 条件查询：使用where子句指定过滤条件
 2. 分组查询：group by子句的使用以及having子句的使用，having子句可以对分组后的记录进行过滤
 3. 排序查询：order by子句的使用，注意如果不指定排序方向，则默认为升序
 4. 内连接查询：inner join的使用
 5. 外连接查询：left join的使用以及right join的使用
 6. 子查询：构建临时表
2. 运算符：
 1. between ... and
 2. in、not in
 3. is null, is not null
 4. intersect（交集）
 5. like
 6. and、or
 7. >、<、!=
 8. union（并集）、union all
3. 更新：update语句的使用
4. 删除：delete语句的使用
5. 聚合函数的使用：max、min、sum、count、avg等。聚合函数通常和分组查询一起使用。
6. 条件表达式：case...when

表定义及表记录如下

1. 表定义

```
create table student(  
    sno int unsigned primary key,  
    sname varchar(32),  
    sex enum('男','女') not null default '男',  
    age tinyint unsigned,  
    sdept varchar(64)  
);  
  
create table sc(  
    sno int unsigned,  
    cno char(4),  
    grade double,  
    primary key(sno,cno)  
);  
  
create table course(  
    cno char(4) primary key,
```

```
    cname varchar(32),
    credit tinyint unsigned,
    semester tinyint unsigned
);
```

2. 表记录

```
insert into student(sno,sname,sex,age,sdept)
values(1512101,'李勇','男',19,'计算机系'),
(1512102,'刘晨','男',20,'计算机系'),
(1512103,'王敏','女',18,'计算机系'),
(1512104,'李小玲','女',19,'计算机系'),
(1521101,'张立','男',22,'信息系'),
(1521102,'吴宾','女',21,'信息系'),
(1521103,'张海','男',20,'信息系'),
(1531101,'钱小平','女',18,'数学系'),
(1531102,'王大力','男',19,'数学系');

insert into sc(sno,cno,grade)
values(1512101,'c001',90),
(1512101,'c002',86),
(1512101,'c003',92),
(1512101,'c005',88),
(1512101,'c006',null),
(1512102,'c001',76),
(1512102,'c002',78),
(1512102,'c005',66),
(1512104,'c002',66),
(1512104,'c005',78),
(1512104,'c008',66),
(1521102,'c001',82),
(1521102,'c005',75),
(1521102,'c007',92),
(1521102,'c009',50),
(1521103,'c002',68),
(1521103,'c006',null),
(1521103,'c007',null),
(1521103,'c008',78),
(1531101,'c001',80),
(1531101,'c005',50),
(1531101,'c007',45),
(1531102,'c001',80),
(1531102,'c002',75),
(1531102,'c005',85),
(1531102,'c009',88);

insert into course(cno,cname,credit,semester)
values('c001','计算机文化学',3,1),
('c002','高等数学',6,1),
('c003','编译原理',3,2),
('c004','大学英语',6,2),
('c005','Java',2,3),
('c006','程序设计',3,3),
('c007','数据结构',5,4),
('c008','操作系统',4,4),
('c009','数据库基础',4,5),
```

```
('c010','计算机网络',5,6);
```

题目

以下题目可以作为编写SQL语句复习之用。

1. 查询学生选课表中的全部数据

```
select * from sc;
```

2. 查询计算机系的学生的姓名、年龄

```
select sname,age from student where sdept = '计算机系';
```

3. 查询成绩在70-80分之间的学生的学号、课程号和成绩

```
select sno,cno,grade from sc where grade between 70 and 80;
```

4. 查询计算机系年龄在18-20之间且性别为“男”的学生的姓名、年龄

```
select sname,age from student where sdept = '计算机系' and sex = '男' and age between 18 and 20;
```

5. 查询课程号为“c001”的课程的最高的分数

```
select max(grade) as '最高分数' from sc where cno = 'c001';
```

6. 查询计算机系学生的最大年龄和最小年龄

```
select max(age) as '最大年龄',min(age) as '最小年龄' from student where sdept = '计算机系';
```

7. 统计每个系的学生个数

```
select count(sno) as '学生人数', sdept from student group by sdept;
```

8. 统计每门课程的选课人数和考试最高分数

```
select count(sno) as '选课人数', max(grade) as '最高分数',cno from sc group by cno;
```

9. 统计每个学生的选课门数和考试总成绩，并按选课门数升序显示结果

```
select count(cno) as select_count,sum(grade),sno from sc group by sno order by select_count;
```

10. 统计各门课程考试成绩的平均分

```
select cno, avg(grade) as '平均分' from sc group by cno;
```

11. 查询总成绩超过200分的学生，要求列出学号、总成绩

```
# 方式1
select sno, sum(grade) as sum_grade from sc group by sno having sum_grade > 200;
# 方式2: 使用内连接
select a.sno, sum(b.grade) as sum_grade from sc b join student a on a.sno = b.sno
group by b.sno having sum_grade > 200;
```

12. 查询选修了"c002"号课程的学生们的姓名和所在系

```
# 方式1
select a.sname, a.sdept from student a inner join sc b on a.sno = b.sno where
b.cno = 'c002';
# 方式2: 使用子查询
select a.sname, a.sdept from student a inner join sc b on a.sno = b.sno where
b.cno in (select cno from sc where cno = 'c002');
```

13. 查询成绩80分以上的学生的姓名、课程号和成绩，并按成绩降序排列结果

```
select a.sname, b.cno, b.grade from sc b inner join student a on a.sno = b.sno
where b.grade > 80 order by b.grade desc;
```

14. 查询哪些学生没有选课，要求列出学号、姓名和所在系

```
select a.sno, a.sname, a.sdept from student a left join sc b on a.sno = b.sno
where b.cno is null;
```

15. 查询与Java在同一学期开设的课程名和开课学期

```
select cname, semester from course where semester in (select semester from course
where cname = 'java') and cname != 'java';
```

16. 查询与李勇年龄相同的学生的姓名、所在系和年龄

```
select sname, sdept, age from student where age in (select age from student where
sname = '李勇') and sname != '李勇';
```

17. 查询数学系成绩80分以上的学生的学号、姓名、课程号和成绩

```
select a.sno, a.sname, b.cno, b.grade from sc b inner join student a on a.sno =
b.sno where a.sdept = '数学系' and b.grade > 80;
```

18. 查询计算机系考试成绩最高的学生的姓名。

```
select a.sname from sc b inner join student a on b.sno = a.sno where a.sno in
(select sno from student where sdept = '计算机系') order by b.grade desc limit 1;
```

19. 查询数据结构考试成绩最高的学生的姓名、所在系、性别和成绩。

```
select a.sname,a.sdept,a.sex,b.grade from sc b inner join student a on a.sno = b.sno inner join course c on b.cno = c.cno where c.cname = '数据结构' order by grade desc limit 1;
```

20. 查询没有选修Java课程的学生的姓名和所在系

```
select a.sname, a.sdept from student a left join sc b on a.sno = b.sno and b.cno in (select cno from course where c.cname = 'java') where b.cno is null;
```

21. 查询计算机系没有选课的学生的姓名和性别

```
select a.sname,a.sex from student a left join sc b on a.sno = b.sno where b.cno is null and a.sdept = '计算机系';
```

22. 删除考试成绩低于50分的学生的选课记录以及没有选课的记录

```
delete from sc where grade < 50;
delete from sc where grade is null;
```

23. 删除没有人选的课程记录

```
delete from course where cno not in (select distinct cno from sc);
```

24. 删除计算机系Java成绩不及格学生的Java选课记录

```
delete from sc where sno in (select sno from student where sdept = '计算机系') and grade < 60 and cno in (select cno from course where c.cname = 'java');
```

25. 将第2学期开设的所有课程的学分增加2分

```
update course set credit = credit + 2 where semester = 2;
```

26. 将Java课程的学分改为3分

```
update course set credit = 3 where c.cname = 'Java';
```

27. 将计算机系学生的年龄增加1岁

```
update student set age = age + 1 where sdept = '计算机系';
```

28. 将信息系学生的“计算机文化学”课程的考试成绩加5分

```
update sc set grade = grade + 5 where sno in (select sno from student where sdept = '信息系') and cno in (select cno from course where c.cname = '计算机文化学');
```

29. 查询每个系年龄大于等于20的学生人数，并将结果保存到一个新永久表dept_age中

```
create table dept_age as select count(*) as '人数' from student where age > 20 group by sdept;
```

30. 查询计算机系每个学生的Java考试情况，列出学号、姓名、成绩和成绩情况，其中成绩情况的显示规则为:如果成绩大于等于90，则成绩情况为“好”；如果成绩在80-89，则成绩情况为“较好”；如果成绩在70-79，则成绩情况为“一般”；如果成绩在60-69，则成绩情况为“较差”；如果成绩小于60，则成绩情况为“差”。

```
SELECT student.sno AS '学号', student.sname AS '姓名', sc.grade AS '成绩',
CASE
WHEN sc.grade >= 90 THEN '好'
WHEN sc.grade BETWEEN 80 AND 89 THEN '较好'
WHEN sc.grade BETWEEN 70 AND 79 THEN '一般'
WHEN sc.grade BETWEEN 60 AND 69 THEN '较差'
ELSE '差'
END AS '成绩情况'
FROM sc
JOIN student ON sc.sno = student.sno
JOIN course ON sc.cno = course.cno
WHERE student.sdept = '计算机系' AND course.cname = 'Java';
```

31. 统计每个学生的选课门数(包括没有选课的学生)，列出学号、选课门数和选课情况，其中选课情况显示规则为:如果选课门数大于等于6门，则选课情况为“多”；如果选课门数超过在3-5门，则选课情况为“一般”；如果选课门数在1-2门，则选课情况为“偏少”；如果没有选课，则选课情况为“未选课”。

```
SELECT a.sno, count(b.cno) as '选课门数',
CASE
WHEN count(b.cno) >= 6 THEN '多'
WHEN count(b.cno) BETWEEN 3 AND 5 THEN '一般'
WHEN count(b.cno) BETWEEN 1 AND 2 THEN '少'
ELSE
'未选课'
END AS '选课情况'
FROM student a LEFT JOIN sc b ON a.sno = b.sno GROUP BY a.sno;
```

32. 修改全部课程的学分，修改规则如下:如果是第1-2学期开设的课程，则学分增加5分;如果是第3-4学期开设的课程，则学分增加3分;如果是第5-6学期开设的课程，则学分增加1分;对其他学期开设的课程，学分不变。

```
update course set credit = credit +
CASE
WHEN semester BETWEEN 1 AND 2 THEN 5
WHEN semester BETWEEN 3 AND 4 THEN 3
WHEN semester BETWEEN 5 AND 6 THEN 1
ELSE
0
END;
```

33. 查询“李勇”和“王大力”所选的全部课程，列出课程名、开课学期和学分，不包括重复的结果。

```
# 方式1: 使用子查询
select cname,semester,credit from course where cno in (select cno from sc INNER
JOIN student on sc.sno = student.sno
where student.sname = '李勇' UNION select cno from sc INNER JOIN student on
sc.sno = student.sno
where student.sname = '王大力' );
# 方式2: 推荐

select distinct c.cname,c.semester,c.credit from sc a
inner join student b on a.sno = b.sno
inner join course c on a.cno = c.cno
where b.sname = '李勇' or b.sname = '王大力';
```

34. 查询在第3学期开设的课程中，“李勇”选了但“王大力”没选的课程，列出课程名和学分。

```
# minus可以求解差集，但是MySQL不支持这个运算符
# 这里使用到了子查询，如果有更好的答案欢迎指出！

select cname, credit from course where cno in
(select b.cno from sc b inner join student a on b.sno = a.sno where a.sname = '李
勇' and b.cno not in (select c.cno from sc c inner join student a on c.sno =
a.sno where a.sname = '王大力'));
```

35. 查询在学分大于等于3分的课程中，“李勇”和“王大力”所选的相同课程，列出课程名和学分。

```
SELECT course.cname, course.credit
FROM course
WHERE cno IN(SELECT cno FROM sc JOIN student ON sc.sno = student.sno WHERE
student.sname = '李勇' INTERSECT SELECT cno FROM sc JOIN student ON student.sno =
sc.sno WHERE student.sname = '王大力')
AND course.credit >= 3;
```

参考: [参考](#)

本文转自 <https://www.cnblogs.com/xiaocer/p/17744423.html>，如有侵权，请联系删除。