表结构：  
CREATE TABLE STUDENT  
(SNO INT NOT NULL PRIMARY KEY,   
SNAME VARCHAR(4) NOT NULL,  
SSEX VARCHAR(2) NOT NULL,   
SBIRTHDAY DATETIME,  
CLASS VARCHAR(5))  
go  
CREATE TABLE COURSE  
(CNO VARCHAR(5) NOT NULL PRIMARY KEY,   
CNAME VARCHAR(10) NOT NULL,   
TNO VARCHAR(10) NOT NULL)  
go  
CREATE TABLE SCORE   
(SNO INT NOT NULL,   
CNO VARCHAR(5) NOT NULL,   
DEGREE NUMERIC(10, 1) NOT NULL,  
FOREIGN KEY (SNO) REFERENCES STUDENT(SNO),  
FOREIGN KEY(CNO) REFERENCES COURSE(CNO)  
)   
go  
CREATE TABLE TEACHER   
(TNO INT NOT NULL,   
TNAME VARCHAR(4) NOT NULL,   
TSEX VARCHAR(2) NOT NULL,   
TBIRTHDAY DATETIME NOT NULL,   
PROF VARCHAR(6),   
DEPART VARCHAR(10) NOT NULL)  
  
INSERT INTO STUDENT (SNO,SNAME,SSEX,SBIRTHDAY,CLASS) VALUES (108 ,'曾华' ,'男' , '1977-09-01', '95033');  
INSERT INTO STUDENT (SNO,SNAME,SSEX,SBIRTHDAY,CLASS) VALUES (105 ,'匡明' ,'男' , '1975-10-02', '95031');  
INSERT INTO STUDENT (SNO,SNAME,SSEX,SBIRTHDAY,CLASS) VALUES (107 ,'王丽' ,'女' , '1976-01-23', '95033');  
INSERT INTO STUDENT (SNO,SNAME,SSEX,SBIRTHDAY,CLASS) VALUES (101 ,'李军' ,'男' , '1976-02-20', '95033');  
INSERT INTO STUDENT (SNO,SNAME,SSEX,SBIRTHDAY,CLASS) VALUES (109 ,'王芳' ,'女' , '1975-02-10', '95031');  
INSERT INTO STUDENT (SNO,SNAME,SSEX,SBIRTHDAY,CLASS) VALUES (103 ,'陆君' ,'男' , '1974-06-03', '95031');  
GO  
INSERT INTO COURSE(CNO,CNAME,TNO)VALUES ('3-105' ,'计算机导论', '825');  
INSERT INTO COURSE(CNO,CNAME,TNO)VALUES ('3-245' ,'操作系统' , '804');  
INSERT INTO COURSE(CNO,CNAME,TNO)VALUES ('6-166' ,'数据电路' , '856');  
INSERT INTO COURSE(CNO,CNAME,TNO)VALUES ('9-888' ,'高等数学' , '100');  
GO  
INSERT INTO SCORE(SNO,CNO,DEGREE)VALUES (103,'3-245',86);  
INSERT INTO SCORE(SNO,CNO,DEGREE)VALUES (105,'3-245',75);  
INSERT INTO SCORE(SNO,CNO,DEGREE)VALUES (109,'3-245',68);  
INSERT INTO SCORE(SNO,CNO,DEGREE)VALUES (103,'3-105',92);  
INSERT INTO SCORE(SNO,CNO,DEGREE)VALUES (105,'3-105',88);  
INSERT INTO SCORE(SNO,CNO,DEGREE)VALUES (109,'3-105',76);  
INSERT INTO SCORE(SNO,CNO,DEGREE)VALUES (101,'3-105',64);  
INSERT INTO SCORE(SNO,CNO,DEGREE)VALUES (107,'3-105',91);  
INSERT INTO SCORE(SNO,CNO,DEGREE)VALUES (108,'3-105',78);  
INSERT INTO SCORE(SNO,CNO,DEGREE)VALUES (101,'6-166',85);  
INSERT INTO SCORE(SNO,CNO,DEGREE)VALUES (107,'6-106',79);  
INSERT INTO SCORE(SNO,CNO,DEGREE)VALUES (108,'6-166',81);  
GO  
INSERT INTO TEACHER(TNO,TNAME,TSEX,TBIRTHDAY,PROF,DEPART)   
VALUES (804,'李诚','男','1958-12-02','副教授','计算机系');  
INSERT INTO TEACHER(TNO,TNAME,TSEX,TBIRTHDAY,PROF,DEPART)   
VALUES (856,'张旭','男','1969-03-12','讲师','电子工程系');  
INSERT INTO TEACHER(TNO,TNAME,TSEX,TBIRTHDAY,PROF,DEPART)  
VALUES (825,'王萍','女','1972-05-05','助教','计算机系');  
INSERT INTO TEACHER(TNO,TNAME,TSEX,TBIRTHDAY,PROF,DEPART)   
VALUES (831,'刘冰','女','1977-08-14','助教','电子工程系');  
  
题目：  
1、 查询Student表中的所有记录的Sname、Ssex和Class列。  
2、 查询教师所有的单位即不重复的Depart列。  
3、 查询Student表的所有记录。  
4、 查询Score表中成绩在60到80之间的所有记录。  
5、 查询Score表中成绩为85，86或88的记录。  
6、 查询Student表中“95031”班或性别为“女”的同学记录。  
7、 以Class降序查询Student表的所有记录。  
8、 以Cno升序、Degree降序查询Score表的所有记录。  
9、 查询“95031”班的学生人数。  
10、查询Score表中的最高分的学生学号和课程号。  
11、查询‘3-105’号课程的平均分。  
12、查询Score表中至少有5名学生选修的并以3开头的课程的平均分数。  
13、查询最低分大于70，最高分小于90的Sno列。  
14、查询所有学生的Sname、Cno和Degree列。  
15、查询所有学生的Sno、Cname和Degree列。  
16、查询所有学生的Sname、Cname和Degree列。  
17、查询“95033”班所选课程的平均分。  
18、假设使用如下命令建立了一个grade表：  
create table grade(low   number(3,0),upp   number(3),rank   char(1));  
insert into grade values(90,100,’A’);  
insert into grade values(80,89,’B’);  
insert into grade values(70,79,’C’);  
insert into grade values(60,69,’D’);  
insert into grade values(0,59,’E’);  
commit;  
现查询所有同学的Sno、Cno和rank列。  
19、查询选修“3-105”课程的成绩高于“109”号同学成绩的所有同学的记录。  
20、查询score中选学一门以上课程的同学中分数为非最高分成绩的记录。  
21、查询成绩高于学号为“109”、课程号为“3-105”的成绩的所有记录。  
22、查询和学号为108的同学同年出生的所有学生的Sno、Sname和Sbirthday列。  
23、查询“张旭“教师任课的学生成绩。  
24、查询选修某课程的同学人数多于5人的教师姓名。  
25、查询95033班和95031班全体学生的记录。  
26、查询存在有85分以上成绩的课程Cno.  
27、查询出“计算机系“教师所教课程的成绩表。  
28、查询“计算机系”与“电子工程系“不同职称的教师的Tname和Prof。  
29、查询选修编号为“3-105“课程且成绩至少高于选修编号为“3-245”的同学的Cno、Sno和Degree,并按Degree从高到低次序排序。  
30、查询选修编号为“3-105”且成绩高于选修编号为“3-245”课程的同学的Cno、Sno和Degree.  
31、查询所有教师和同学的name、sex和birthday.  
32、查询所有“女”教师和“女”同学的name、sex和birthday.  
33、查询成绩比该课程平均成绩低的同学的成绩表。  
34、查询所有任课教师的Tname和Depart.  
35  查询所有未讲课的教师的Tname和Depart.   
36、查询至少有2名男生的班号。  
37、查询Student表中不姓“王”的同学记录。  
38、查询Student表中每个学生的姓名和年龄。  
39、查询Student表中最大和最小的Sbirthday日期值。  
40、以班号和年龄从大到小的顺序查询Student表中的全部记录。  
41、查询“男”教师及其所上的课程。  
42、查询最高分同学的Sno、Cno和Degree列。  
43、查询和“李军”同性别的所有同学的Sname.  
44、查询和“李军”同性别并同班的同学Sname.  
45、查询所有选修“计算机导论”课程的“男”同学的成绩表  
参考答案：  
1. SELECT SNAME,SSEX,CLASS FROM STUDENT;  
  
2. SELECT DISTINCT DEPART FROM TEACHER;  
  
3. SELECT \* FROM STUDENT;  
  
4. SELECT \* FROM SCORE WHERE DEGREE BETWEEN 60 AND 80;  
  
5.SELECT \* FROM SCORE WHERE DEGREE IN (85,86,88);  
  
6. SELECT \* FROM STUDENT WHERE CLASS='95031' OR SSEX='女';  
  
7.SELECT \* FROM STUDENT ORDER BY CLASS DESC;  
  
8.SELECT \* FROM SCORE ORDER BY CNO ASC,DEGREE DESC;  
  
9.SELECT  COUNT(\*) FROM STUDENT WHERE CLASS='95031';  
  
10.SELECT SNO,CNO FROM SCORE WHERE DEGREE=(SELECT MAX(DEGREE) FROM SCORE);  
  
SELECT SNO,CNO FROM SCORE ORDER BY DEGREE DESC LIMIT 1;  
  
11.SELECT AVG(DEGREE) FROM SCORE WHERE CNO='3-105';  
  
12.select avg(degree),cno  
from score  
where cno like '3%'  
group by cno  
having count(sno)>= 5;  
  
13.SELECT SNO FROM SCORE GROUP BY SNO HAVING MIN(DEGREE)>70 AND MAX(DEGREE)<90;  
  
14.SELECT A.SNAME,B.CNO,B.DEGREE FROM STUDENT AS A JOIN SCORE AS B ON A.SNO=B.SNO;  
  
15.SELECT A.CNAME, B.SNO,B.DEGREE FROM COURSE AS A JOIN SCORE AS B ON A.CNO=B.CNO ;  
  
16.SELECT A.SNAME,B.CNAME,C.DEGREE FROM STUDENT A JOIN (COURSE B,SCORE C)  
ON A.SNO=C.SNO AND B.CNO =C.CNO;  
  
17.SELECT AVG(A.DEGREE) FROM SCORE A JOIN STUDENT B ON A.SNO = B.SNO WHERE B.CLASS='95033';  
  
18.SELECT A.SNO,A.CNO,B.RANK FROM SCORE A,GRADE B WHERE A.DEGREE BETWEEN B.LOW AND B.UPP   
  
ORDER BY RANK;  
  
19.SELECT A.\* FROM SCORE A JOIN SCORE B WHERE A.CNO='3-105' AND A.DEGREE>B.DEGREE AND   
  
B.SNO='109' AND B.CNO='3-105';  
另一解法：SELECT A.\* FROM SCORE A  WHERE A.CNO='3-105' AND A.DEGREE>ALL(SELECT DEGREE FROM   
  
SCORE B WHERE B.SNO='109' AND B.CNO='3-105');  
  
20.SELECT \* FROM score s WHERE DEGREE<(SELECT MAX(DEGREE) FROM SCORE) GROUP BY SNO HAVING   
  
COUNT(SNO)>1 ORDER BY DEGREE ;  
  
21.见19的第二种解法  
  
22。SELECT SNO,SNAME,SBIRTHDAY FROM STUDENT WHERE YEAR(SBIRTHDAY)=(SELECT YEAR(SBIRTHDAY)   
  
FROM STUDENT WHERE SNO='108');  
ORACLE:select x.cno,x.Sno,x.degree from score x,score y where x.degree>y.degree and   
  
y.sno='109'and y.cno='3-105';  
select cno,sno,degree from score   where degree >(select degree from score where sno='109'   
  
and cno='3-105')  
  
23.SELECT A.SNO,A.DEGREE FROM SCORE A JOIN (TEACHER B,COURSE C)  
ON A.CNO=C.CNO AND B.TNO=C.TNO  
WHERE B.TNAME='张旭';  
另一种解法：select cno,sno,degree from score where cno=(select x.cno from course x,teacher y   
  
where x.tno=y.tno and y.tname='张旭');  
根据实际EXPLAIN此SELECT语句，第一个的扫描次数要小于第二个  
  
24.SELECT A.TNAME FROM TEACHER A JOIN (COURSE B, SCORE C) ON (A.TNO=B.TNO AND B.CNO=C.CNO)   
  
GROUP BY C.CNO HAVING COUNT(C.CNO)>5;  
另一种解法：select tname from teacher where tno in(select x.tno from course x,score y where   
  
x.cno=y.cno group by x.tno having count(x.tno)>5);  
实际测试1明显优于2  
  
  
25。select cno,sno,degree from score where cno=(select x.cno from course x,teacher y where   
  
x.tno=y.tno and y.tname='张旭');  
  
26。SELECT CNO FROM SCORE GROUP BY CNO HAVING MAX(DEGREE)>85;  
另一种解法：select distinct cno from score where degree in (select degree from score where   
  
degree>85);  
  
27。SELECT A.\* FROM SCORE A JOIN (TEACHER B,COURSE C) ON A.CNO=C.CNO AND B.TNO=C.TNO  
WHERE B.DEPART='计算机系';  
另一种解法：SELECT \* from score where cno in (select a.cno from course a join teacher b on   
  
a.tno=b.tno and b.depart='计算机系');  
此时2略好于1，在多连接的境况下性能会迅速下降  
  
28。select tname,prof from teacher where depart='计算机系' and prof not in (select prof from   
  
teacher where depart='电子工程系');  
  
29。SELECT \* FROM SCORE WHERE DEGREE>ANY(SELECT DEGREE FROM SCORE WHERE CNO='3-245') ORDER   
  
BY DEGREE DESC;  
  
30。SELECT \* FROM SCORE WHERE DEGREE>ALL(SELECT DEGREE FROM SCORE WHERE CNO='3-245') ORDER   
  
BY DEGREE DESC;  
  
31.SELECT SNAME AS NAME, SSEX AS SEX, SBIRTHDAY AS BIRTHDAY FROM STUDENT  
UNION  
SELECT TNAME AS NAME, TSEX AS SEX, TBIRTHDAY AS BIRTHDAY FROM TEACHER;  
  
32.SELECT SNAME AS NAME, SSEX AS SEX, SBIRTHDAY AS BIRTHDAY FROM STUDENT WHERE SSEX='女'  
UNION  
SELECT TNAME AS NAME, TSEX AS SEX, TBIRTHDAY AS BIRTHDAY FROM TEACHER WHERE TSEX='女';  
  
33.SELECT A.\* FROM SCORE A WHERE DEGREE<(SELECT AVG(DEGREE) FROM SCORE B WHERE A.CNO=B.CNO);  
须注意\*\*\*\*\*\*\*\*此题  
  
34。解法一：SELECT A.TNAME,A.DEPART FROM TEACHER A JOIN COURSE B ON A.TNO=B.TNO;  
解法二：select tname,depart from teacher a where exists  
(select \* from course b where a.tno=b.tno);  
解法三：SELECT TNAME,DEPART FROM TEACHER WHERE TNO IN (SELECT TNO FROM COURSE);  
  
实际分析，第一种揭发貌似更好，至少扫描次数最少。  
  
35.解法一：SELECT TNAME,DEPART FROM TEACHER A LEFT JOIN COURSE B USING(TNO) WHERE ISNUL  
  
(B.tno);  
解法二：select tname,depart from teacher a where not exists  
(select \* from course b where a.tno=b.tno);  
解法三：SELECT TNAME,DEPART FROM TEACHER WHERE TNO NOT IN (SELECT TNO FROM COURSE);  
NOT IN的方法效率最差，其余两种差不多  
  
36.SELECT CLASS FROM STUDENT A WHERE SSEX='男' GROUP BY CLASS HAVING COUNT(SSEX)>1;  
  
37.SELECT \* FROM STUDENT A WHERE SNAME not like '王%';  
  
38.SELECT SNAME,(YEAR(NOW())-YEAR(SBIRTHDAY)) AS AGE FROM STUDENT;  
  
39.select sname,sbirthday as THEMAX from student where sbirthday =(select min(SBIRTHDAY)   
  
from student)  
union  
select sname,sbirthday as THEMIN from student where sbirthday =(select max(SBIRTHDAY) from   
  
student);  
  
40.SELECT CLASS,(YEAR(NOW())-YEAR(SBIRTHDAY)) AS AGE FROM STUDENT ORDER BY CLASS DESC,AGE   
  
DESC;  
  
41.SELECT A.TNAME,B.CNAME FROM TEACHER A JOIN COURSE B USING(TNO) WHERE A.TSEX='男';  
  
42.SELECT A.\* FROM SCORE A WHERE DEGREE=(SELECT MAX(DEGREE) FROM SCORE B );  
  
43.SELECT SNAME FROM STUDENT A WHERE SSEX=(SELECT SSEX FROM STUDENT B WHERE B.SNAME='李军');  
  
44.SELECT SNAME FROM STUDENT A WHERE SSEX=(SELECT SSEX FROM STUDENT B WHERE B.SNAME='李军' )  
AND CLASS=(SELECT CLASS FROM STUDENT C WHERE c.SNAME='李军');  
  
45.解法一：SELECT A.\* FROM SCORE A JOIN (STUDENT B,COURSE C) USING(sno,CNO) WHERE B.SSEX='男  
  
' AND C.CNAME='计算机导论';  
解法二：select \* from score where sno in(select sno from student where  
ssex='男') and cno=(select cno from course  
where cname='计算机导论');