第一题

SELECT count(CHOICES.score)

FROM COURSES,CHOICES

WHERE COURSES.cname='c++' and COURSES.cid=CHOICES.cid and CHOICES.score>=60

SELECT count(CHOICES.score)

FROM COURSES,CHOICES

WHERE COURSES.cname='c++' and COURSES.cid=CHOICES.cid and CHOICES.score<60

select COUNT(\*)

FROM COURSES,CHOICES

WHERE COURSES.cname='c++' and COURSES.cid=CHOICES.cid

select COUNT(\*)

FROM COURSES,CHOICES

WHERE COURSES.cname='c++' and COURSES.cid=CHOICES.cid and CHOICES.score is NULL

会发现前两个查询的结果加在一起并不等于第三个查询的结果，是因为score为NULL值的被忽略了，没有被统计，第四个查询和前两个查询的和就等于第三个查询，可见NULL与所有比较运算符都是不匹配的。

第二题

SELECT CHOICES.sid,CHOICES.score

FROM COURSES,CHOICES

WHERE COURSES.cname='c++' and COURSES.cid=CHOICES.cid

ORDER BY score ASC

存在NULL值，如果用ASC顺序排序的话就会出现在结果的开头部分

第三题

SELECT distinct CHOICES.score

FROM CHOICES

WHERE cid in (

select COURSES.cid from COURSES where COURSES.cname='c++' and COURSES.cid=CHOICES.cid)

ORDER BY score ASC

如果在第二题的基础上直接加上DISTINCT后NULL不会聚集，因为DISTINCT是同时作用于分数和id的，要两个都完全相同才会去重，如果要NULL聚集就把取学号的去掉。会发现所有的NULL被当作相同的去重复了最后只剩下一个NULL

第四题

select grade

from STUDENTS

GROUP BY grade

会得到15个组，但存在NULL的空值，只有14个年级的分组有意义

第五题

select cid,AVG(score) AVG,COUNT(\*) NUM,MAX(score)MaxS,MIN(score) MinS

from CHOICES

group by cid

平均分，最高分最低分都可以输出值，可见group的过程中忽略了空值，将count函数中的\*变成score，可以看到第三列的值减少了，可见课程中存在空值，在count(\*)中没有被忽略，而改成score后就被忽略了，因此数量变少了。

第六题

select distinct grade

from STUDENTS as s1

where s1.grade>=all(

select s2.grade

from STUDENTS as s2

where s2.grade is not NULL)

如果嵌套的子查询中存在NULL，那么s1.grade>=all的结果就是假的，因此查询到空的,所以需要提前把NULL的值去掉才能查询到结果