预定了所有白色船的水手姓名

因为数据集，我们改成蓝色船

思路：首先我们要查到所有白色船的数量，然后查出所有用户预定的白色船的数量，只要有一个用户预定的白色船的数量和所有白色船的数量相等就好。

select sailorsX.sname from sailors sailorsX

where

(select count(distinct boats.bid) from sailors, boats, reserves where sailors.sid = sailorsX.sid and sailors.sid=reserves.sid and boats.bid=reserves.bid and boats.color=‘blue')

=

(select count(\*) from boats where boats.color='blue');

我们首先在sailors这个表中找sname，这个记录对应的人应该是订阅的蓝色船的数量在去重之后的数量应该是和所有蓝色船的数量是相同的。这样子就可以得出是不是预定了所有蓝色船。

预定最多次的水手的编号

这个部分我们拆分成几个子问题，首先我们需要知道预定次数最多是多少次，然后我们遍历每一个sid，看看这个这个sid会出现几次，然后我们看看这个出现的次数是不是最多的预定次数。

select distinct sid from reserves reservesX where

(select count(\*) from reserves where reservesX.sid = sid )

=

(select MAX(boatcount) from

(select sid , count(\*) as boatcount from reserves group by sid)tmp

);

值的一提的是分组的东西必须在必须在select中出现，反之没有group by的东西不能出现在select后面。比如tmp表的子查询就不能出现select bid这个东西，因为group by后面的属性没有。等于是在group by的语句的同级select中只能出现group by后面的属性和聚合函数。

查询每个水手的编号姓名和预定船的平均次数

没懂平均次数的意思是什么，我们这里就按照次数来看吧。

select sailors.sid , sailors.sname , count(\*) from sailors , reserves where sailors.sid = reserves.sid group by sid , sname;

这里要注意的是在连接的过程中相同的属性可能会合并。所以group by后面直接加属性名即可。

预定了仅仅一次205号船的水手的姓名

select sname from sailors sailorsX where (select count(\*) from reserves where sailorsX.sid = reserves.sid and reserves.bid = 105) = 1;

每个级别水手预定蓝色船的最大数目

这个过程需要分成两步，我们需要知道每个水手预定蓝色船的数目，然后我们用这个中间结论将水手按等级分组，在组中我们使用max就可以得到每个组预定蓝色船的最大值。

select rating, max(countBoats) from

(

select sailors.sid , sailors.rating , count(\*) countBoats from sailors , reserves , boats where sailors.sid=reserves.sid and boats.bid=reserves.bid and boats.color='blue' group by sid,rating

)tmp

group by rating;

2014年次数最多的客人的姓名和和入住次数

With tmp as

(

select 身份证 , 名字 , count (\*) 入住次数 from 用户，房间 ，住宿 where 用户.身份证号=住宿.身份证号 and 房间.房间号=住宿.房间号 and 住宿.入住时间 = 2014 group by 身份证

)

select tmp.用户姓名 , tmp.入住次数 from tmp where 入住次数 =

(

select max(入住次数) from tmp;

)

设教学数据库Education有三个关系：

学生关系S（SNO，SNAME，AGE，SEX，SDEPT）；学习关系SC（SNO，CNO，GRADE）；课程关系C（CNO，CNAME，CDEPT，TNAME）

查询问题：

1. 检索计算机系的全体学生的学号，姓名和性别；
2. 检索学习课程号为C2的学生学号与姓名；
3. 检索选修课程名为“DS”的学生学号与姓名；

（4）检索选修课程号为C2或C4的学生学号；

（5）检索至少选修课程号为C2和C4的学生学号；

（6）检索不学C2课的学生姓名和年龄；

（7）检索学习全部课程的学生姓名；

（8）查询所学课程包含学生S3所学课程的学生学号。

（1）检索计算机系的全体学生的学号，姓名和性别；

SELECT Sno，Sname，Sex

FROM S

WHERE Sdept =’CS’；

（2）检索学习课程号为C2的学生学号与姓名；

2.SELECT S.Sno，Sname

FROM S，SC

WHERE S.Sno=SC.Sno

AND SC.Cno=‘C2’;

1.SELECT Sno，Sname

FROM S

WHERE Sno IN

（ SELECT Sno

FROM SC

WHERE Cno=‘C2’)

（3）检索选修课程名为“DS”的学生学号与姓名

本查询涉及到学号、姓名和课程名三个属性，分别存放在S和C表中，但S和C表没有直接联系，必须通过SC表建立它们二者的联系。 C → SC → S

基本思路：

（1）首先在C表中找出“DS”课程的课程号Cno；

（2）然后在SC表中找出Cno等于第一步给出的Cno 集合中的某个元素Cno；

（3）最后在S关系中选出Sno等于第二步中Sno 集合中某个元素的元组，取出Sno 和Sname送入结果表列。

SELECT Sno，Sname

FROM S

WHERE Sno IN

（SELECT Sno

FROM SC

WHERE Cno IN

（SELECT Cno

FROM C

WHERE Cname=‘DS’））；

1. 检索选修课程号为C2或C4的学生学号；

SELECT Sno

FROM SC

WHERE Cno=‘C2’ OR Cno=‘C4’；

1. 检索至少选修课程号为C2和C4的学生学号；

SELECT Sno

FROM SC X，SC Y

WHERE X.Sno=Y.Sno AND X.Cno=‘C2’ AND Y.Cno=‘C4’ ；

1. 检索不学C2课的学生姓名和年龄；

2.SELECT Sname

FROM S

WHERE NOT EXISTS

（ SELECT \*

FROM SC

WHERE SC.Sno=S.Sno

AND Cno=‘C2’)；

1.SELECT Sname

FROM S

WHERE Sno NOT IN

（ SELECT Sno

FROM SC

WHERE Cno=‘C2’)；

1. 检索学习全部课程的学生姓名；

在表S中找学生，要求这个学生学了全部课程。换言之，在S表中找学生，在C中不存在一门课程，这个学生没有学。

SELECT Sname

FROM S

WHERE NOT EXISTS

（SELECT \*

FROM C

WHERE NOT EXISTS

（SELECT \*

FROM SC

WHERE SC.Sno=S.Sno AND SC.Cno=C.Cno）)；

1. 查询所学课程包含学生S3所学课程的学生学号。

分析：不存在这样的课程Y，学生S3选了Y，而其他学生没有选。

SELECT DISTINCT Sno

FROM SC AS X

WHERE NOT EXISTS

（SELECT \*

FROM SC AS Y

WHERE Y.Sno=‘S3’ AND NOT EXISTS

（SELECT \*

FROM SC AS Z

WHERE Z.Sno=X.Sno AND Z.Cno=Y.Cno）)；

**设教学数据库Education有三个关系：**

**学生关系S（SNO，SNAME，AGE，SEX，SDEPT）；**

**学习关系SC（SNO，CNO，GRADE）；**

**课程关系C（CNO，CNAME，CDEPT，TNAME）**

**查询问题：**

1：查所有年龄在20岁以下的学生姓名及年龄。

2：查考试成绩有不及格的学生的学号

3：查所年龄在20至23岁之间的学生姓名、系别及年龄。

4：查计算机系、数学系、信息系的学生姓名、性别。

5：查既不是计算机系、数学系、又不是信息系的学生姓名、性别

6：查所有姓“刘”的学生的姓名、学号和性别。

7：查姓“上官”且全名为3个汉字的学生姓名。

8：查所有不姓“张”的学生的姓名。

9：查DB\_Design课程的课程号。

10：查缺考的学生的学号和课程号。

11：查年龄为空值的学生的学号和姓名。

12：查计算机系20岁以下的学生的学号和姓名。

13：查计算机系、数学系、信息系的学生姓名、性别。

14：查询选修了C3课程的学生的学号和成绩，其结果按分数的降序排列。

15：查询全体学生的情况，查询结果按所在系升序排列，对同一系中的学生按年龄降序排列。

16：查询学生总人数。

17：查询选修了课程的学生人数。

18：计算选修了C1课程的学生平均成绩。

19：查询学习C3课程的学生最高分数。

20：查询各个课程号与相应的选课人数。

21：查询计算机系选修了3门以上课程的学生的学号。

22：求基本表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。

23：查询每个学生及其选修课程的情况。

24：查询选修了C2课程且成绩在90分以上的所有学生。

25：查询每个学生选修的课程名及其成绩。

26：统计每一年龄选修课程的学生人数。

27：查询选修了C2课程的学生姓名。

28：查询与“张三”在同一个系学习的学生学号、姓名和系别。

29：查询选修课程名为“数据库”的学生学号和姓名。

30：查询与“张三”在同一个系学习的学生学号、姓名和系别。

31：查询选修课程名为“数据库”的学生学号和姓名。

32：查询选修了C2课程的学生姓名。

33：查询所有未选修C2课程的学生姓名。

34：查询与“张三”在同一个系学习的学生学号、姓名和系别。

35：查询选修了全部课程的学生姓名。

36：查询所学课程包含学生S3所学课程的学生学号

**（1）比较**

**例1**：查所有年龄在20岁以下的学生姓名及年龄。

SELECT Sname，Sage

FROM S

WHERE Sage<20; (NOT age>=20)

**例2：**查考试成绩有不及格的学生的学号

SELECT DISTINCT Sno

FROM SC

WHERE grade<60；

**(2)确定范围**

**例3：**查所年龄在20至23岁之间的学生姓名、系别及年龄。

SELECT Sname，Sdept，Sage

FROM S

WHERE Sage BETWEEN 20 AND 23；

**（3）确定集合**

**例4：**查计算机系、数学系、信息系的学生姓名、性别。

SELECT Sname，Ssex

FROM S

WHERE Sdept IN （’CS’, ‘IS’, ‘MATH’）；

**例5：**查既不是计算机系、数学系、又不是信息系的学生姓名、性别

SELECT Sname，Ssex

FROM S

WHERE Sdept NOT IN （’CS’, ‘IS’, ‘MATH’）；

**（4）字符匹配**

**例6：**查所有姓“刘”的学生的姓名、学号和性别。

SELECT Sname，Sno，Ssex

FROM S

WHERE Sname LIKE ‘刘%’；

**例7：**查姓“上官”且全名为3个汉字的学生姓名。

SELECT Sname

FROM S

WHERE Sname LIKE ‘上官\_ \_’；

**例8：**查所有不姓“张”的学生的姓名。

SELECT Sname，Sno，Ssex

FROM S

WHERE Sname NOT LIKE ‘张%’；

**例9：**查DB\_Design课程的课程号。

SELECT Cno

FROM C

WHERE Cname LIKE ‘DB\\_Design’ ESCAPE ‘\’；

**(5)涉及空值的查询**

**例10**：查缺考的学生的学号和课程号。

SELECT Sno，Cno

FROM SC

WHERE Grade IS NULL； （不能用=代替）

{ 有成绩的 WHERE Grade IS NOT NULLL；}

**例11：**查年龄为空值的学生的学号和姓名。

SELECT Sno，Sname

FROM S

WHERE Sage IS NULL；

**（6）多重条件查询**

**例12：**查计算机系20岁以下的学生的学号和姓名。

SELECT Sno，Sname

FROM S

WHERE Sdept=‘CS’ AND Sage<20;

**例13：**查计算机系、数学系、信息系的学生姓名、性别。

SELECT Sname，Ssex

FROM S

WHERE Sdept =’CS’ OR Sdept =‘IS’ OR Sdept =’MATH’）；

**3、对查询结果排序**

**例14：**查询选修了C3课程的学生的学号和成绩，其结果按分数的降序排列。

SELECT Sno，Grade

FROM SC

WHERE Cno=‘C3’

ORDER BY Grade DESC;

**例15：**查询全体学生的情况，查询结果按所在系升序排列，对同一系中的学生按年龄降序排列。

SELECT \*

FROM S

ORDER BY Sdep，Sage DESC；

**4.聚合函数的使用**

**例16**：查询学生总人数。

SELECT COUNT（\*）

FROM S

例17：查询选修了课程的学生人数。

SELECT COUNT（DISTINCT Sno）

FROM SC

**例18：**计算选修了C1课程的学生平均成绩。

SELECT AVG（Grade）

FROM SC

WHERE Cno=‘C1’；

**例19：**查询学习C3课程的学生最高分数。

SELECT MAX（Grade）

FROM SC

WHERE Cno=‘C3’；

5、对查询结果分组

例20：查询各个课程号与相应的选课人数。

SELECT Cno，COUNT（Sno）

FROM SC

GROUP BY Cno；

该SELECT语句对SC表按Cno的取值进行分组，所有具有相同Cno值的元组为一组，然后对每一组作用聚合函数COUNT以求得该组的学生人数。

如果分组后还要求按一定的条件对这些组进行筛选，最终只输出满足指定条件 组，则可以使用HAVING短语指定筛选条件。

**例21：**查询计算机系选修了3门以上课程的学生的学号。

SELECT Sno

FROM SC

WHERE Sdept=‘CS’

GROUP BY Sno

HAVING COUNT（\*）>3；

WHERE子句与HAVING短语的根本区别在于作用对象不同。WHERE子句作用于基本表或视图，从中选择满足条件的元组。HAVING短语作用于组，从中选择满足条件的组。

**例22**：求基本表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。

SELECT Sage，COUNT（Sno）

FROM S

WHERE Ssex='M'

GROUP BY Sage

HAVING COUNT（\*）> 50

ORDER BY 2，Sage DESC；

**二、多表查询**

**1、联接查询**

**例23：**查询每个学生及其选修课程的情况。

SELECT S.Sno,Sname,Sage,Ssex,Sdept,Cno,Grade

FROM S, SC

WHERE S.Sno=SC.Sno；

**例24：**查询选修了C2课程且成绩在90分以上的所有学生。

SELECT S.Sno，Sname

FROM S，SC

WHERE S.Sno=SC.Sno

AND SC.Cno=‘C2’

AND SC.Grade > 90;

**例25：**查询每个学生选修的课程名及其成绩。

SELECT S.Sno，Sname，Cname，SC.Grade

FROM S，SC，C

WHERE S.Sno=SC.Sno AND SC.Cno=C.Cno

**例26：**统计每一年龄选修课程的学生人数。

SELECT Sage，COUNT（DISTINCT S.Sno）

FROM S，SC

WHERE S.Sno=SC.Sno

GROUP BY S；

由于要统计每一个年龄的学生人数，因此要把满足WHERE子句中条件的查询结果按年龄分组，在每一组中的学生年龄相同。此时的SELECT子句应对每一组分开进行操作，在每一组中，年龄只有一个值，统计的人数是这一组中的学生人数。

**1、嵌套查询**

(1)带有IN谓词的子查询

指父查询与子查询之间用IN进行联接，判断某个属性列值是否在子查询的结果中。

**例27：**查询选修了C2课程的学生姓名。

SELECT Sname

FROM S

WHERE Sno IN

（ SELECT Sno

FROM SC

WHERE Cno=‘C2’);

**例28：**查询与“张三”在同一个系学习的学生学号、

姓名和系别。

分析：

（1）确定“张三”所在的系；

（2）查找所有在X系学习的学生。

SELECT Sdept

FROM S

WHERE Sname=‘张三’；

SELECT Sno，Sname，Sdept

FROM S

WHERE Sdept=‘X’

把第一步查询嵌入到第二步查询中，用以构造第二步查询的条件。

FROM S AS S1， S AS S2

WHERE S1.Sdept=S2.Sdept

AND S2.Sname=‘张三’

SELECT Sno，Sname，Sdept

FROM S

WHERE Sdept IN

（SELECT Sdept

FROM S

WHERE Sname=‘张三’）；

**例29：**查询选修课程名为“数据库”的学生学号和姓名。

本查询涉及到学号、姓名和课程名三个属性，分别存放在S和C表中，但S和C表没有直接联系，必须通过SC表建立它们二者的联系。 C → SC → S

基本思路：

（1）首先在C表中找出“DB”课程的课程号Cno；

（2）然后在SC表中找出Cno等于第一步给出的Cno 集合中的某个元素Cno；

（3）最后在S关系中选出Sno等于第二步中Sno 集合中某个元素的元组，取出Sno 和Sname送入结果表列。

SELECT Sno，Sname

FROM S

WHERE Sno IN

（SELECT Sno

FROM SC

WHERE Cno IN

（SELECT Cno

FROM C

WHERE Cname=‘DB’））；

**联接查询方式**

**(2)带有比较运算符的子查询**

**例30：**查询与“张三”在同一个系学习的学生学号、

姓名和系别。

SELECT Sno，Sname，Sdept

FROM S

WHERE Sdept =

（SELECT Sdept

FROM S

WHERE Sname=‘张三’）；

**例31：**查询选修课程名为“数据库”的学生学号和姓名。

SELECT Sno，Sname

FROM S

WHERE Sno IN

（SELECT Sno

FROM SC

WHERE Cno =

（SELECT Cno

FROM C

WHERE Cname=‘DB’））；

**（3）带有EXISTS谓词的子查询**

（1）带有EXISTS谓词的子查询不返回任何实际数据，它只产生逻辑值。

**例32：**查询选修了C2课程的学生姓名。

1.SELECT Sname

FROM S

WHERE Sno IN

（ SELECT Sno

FROM SC

WHERE Cno=‘C2’);

2.SELECT Sname

FROM S

WHERE EXISTS

（ SELECT \*

FROM SC

WHERE SC.Sno=S.Sno AND Cno=‘C2’);

**例33：**查询所有未选修C2课程的学生姓名。

SELECT Sname

FROM S

WHERE NOT EXISTS

（ SELECT \*

FROM SC

WHERE SC.Sno=S.Sno AND Cno=‘C2’);

[NOT]EXISTS 实际上是一种内、外层互相关的嵌套查询，只有当内层引用了外层的值，这种查询才有意义。

**例34：**查询与“张三”在同一个系学习的学生学号、

姓名和系别。

SELECT Sno，Sname，Sdept

FROM S AS S1

WHERE EXISTS

（SELECT \*

FROM S AS S2

WHERE S2.Sdept=S1. Sdept AND S2.Sname=‘张三’）；

**相关子查询**

**例35：**查询选修了全部课程的学生姓名。

在表S中找学生，要求这个学生学了全部课程。换言之，在S表中找学生，在C中不存在一门课程，这个学生没有学。

SELECT Sname

FROM S

WHERE NOT EXISTS

（SELECT \*

FROM C

WHERE NOT EXISTS

（SELECT \*

FROM SC

WHERE SC.Sno=S.Sno AND SC.Cno=C.Cno）)；

**例36：**查询所学课程包含学生S3所学课程的学生学号

分析：不存在这样的课程Y，学生S3选了Y，而其他学生没有选。

SELECT DISTINCT Sno

FROM SC AS X

WHERE NOT EXISTS

（SELECT \*

FROM SC AS Y

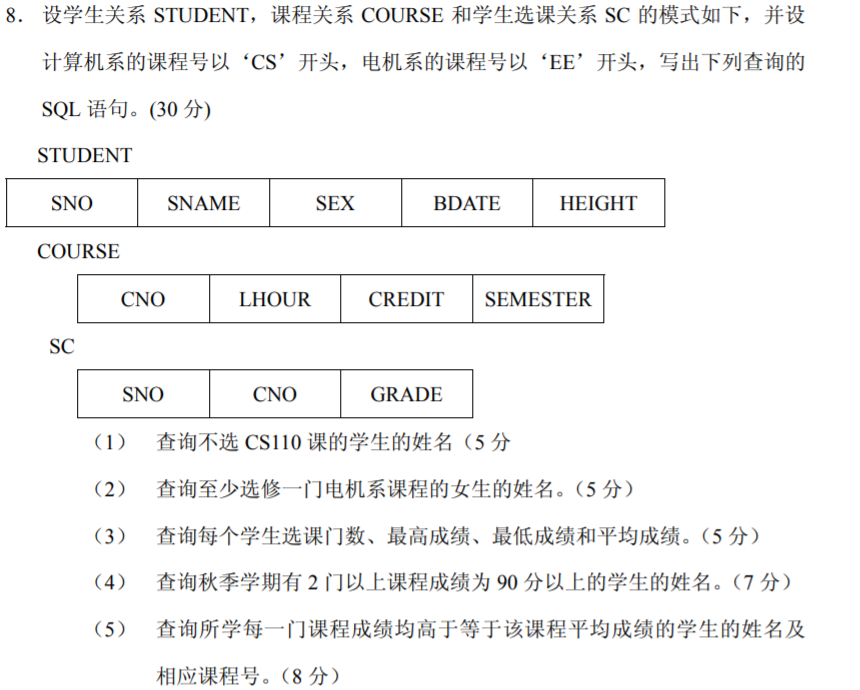
WHERE Y.Sno=‘S3’ AND NOT EXISTS

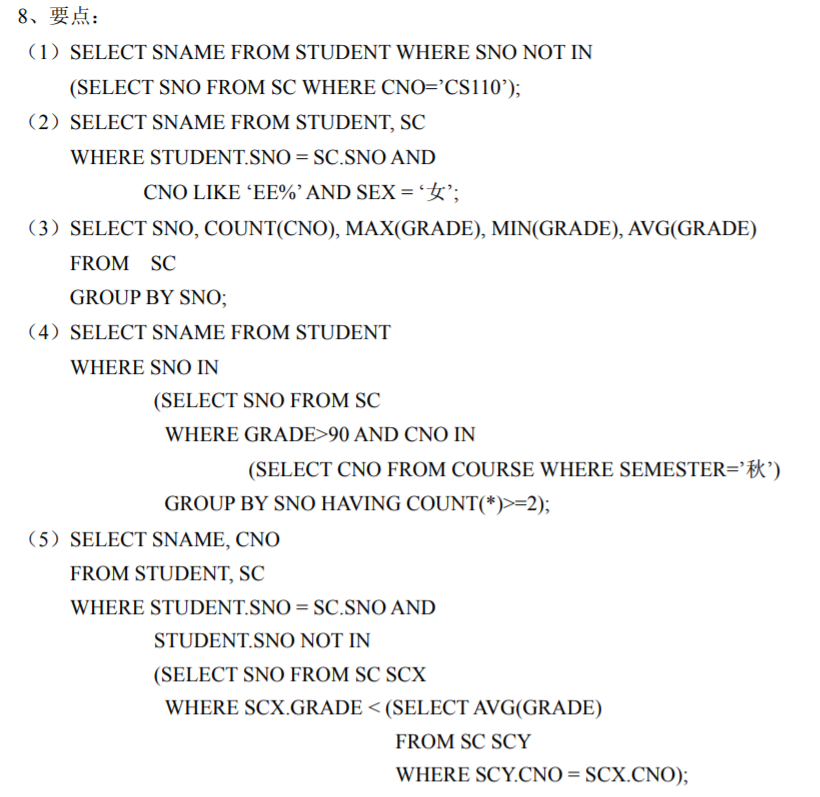
（SELECT \*

FROM SC AS Z

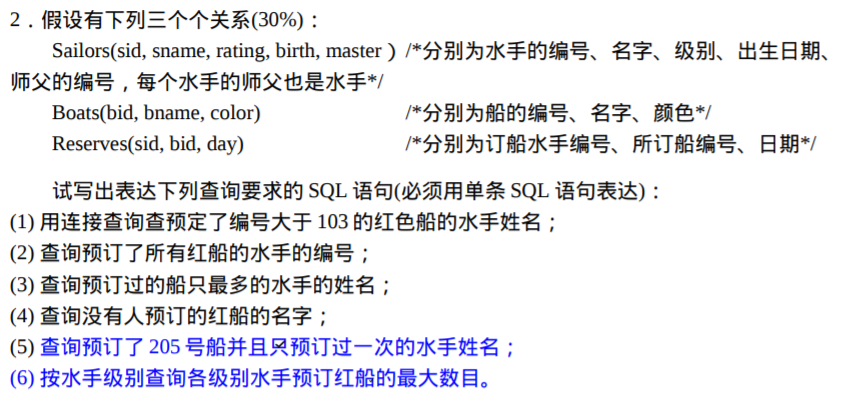
WHERE Z.Sno=X.Sno AND Z.Cno=Y.Cno）)；

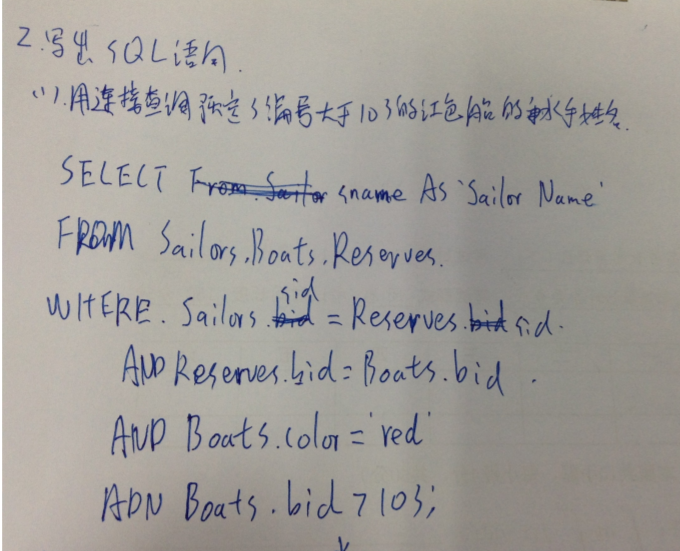
考试1

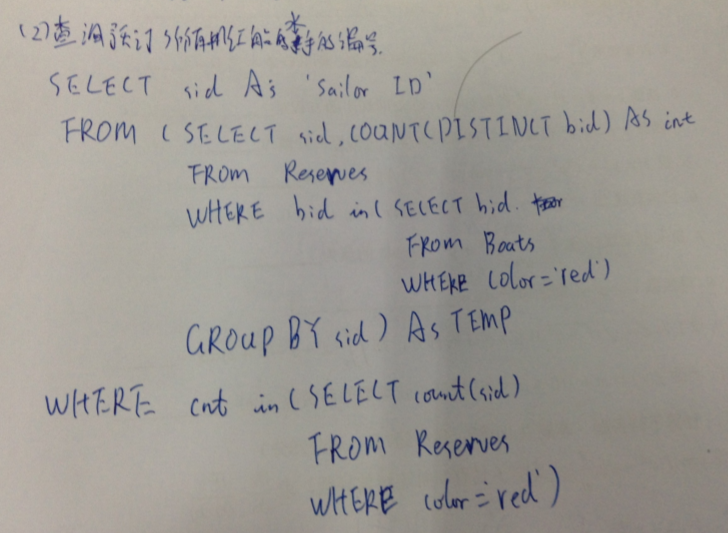


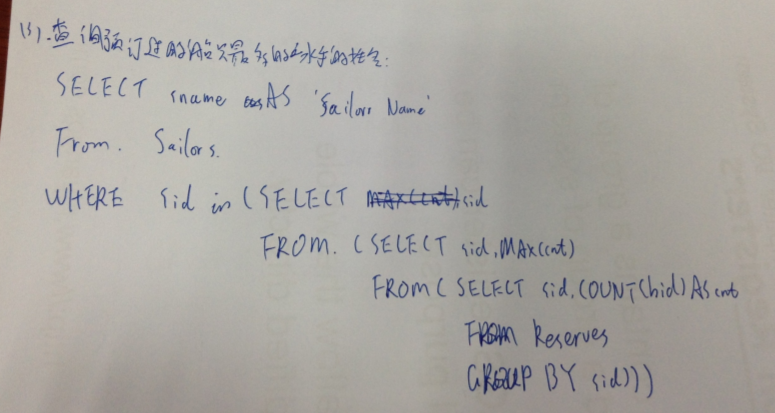


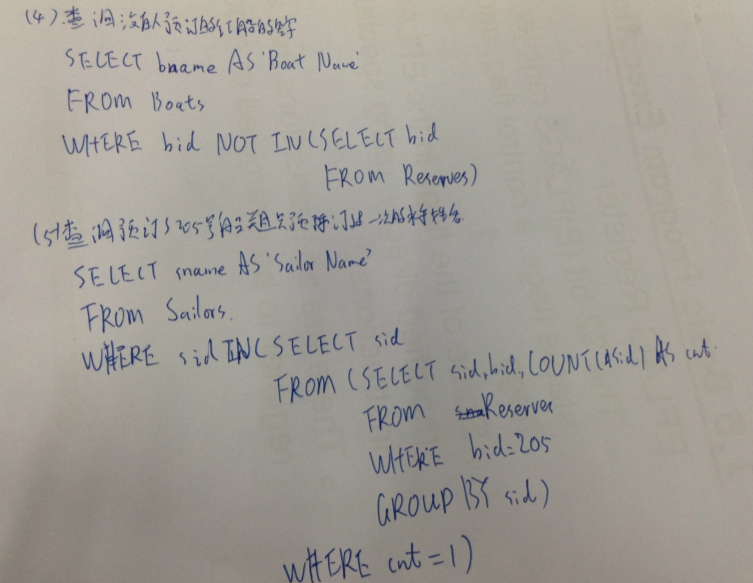
考试2

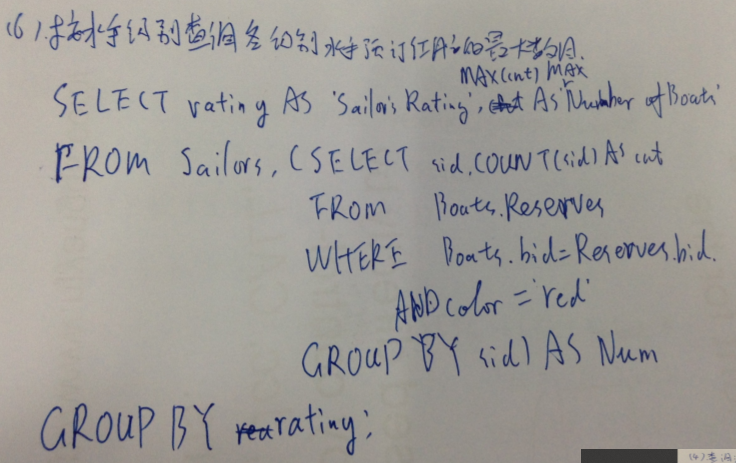




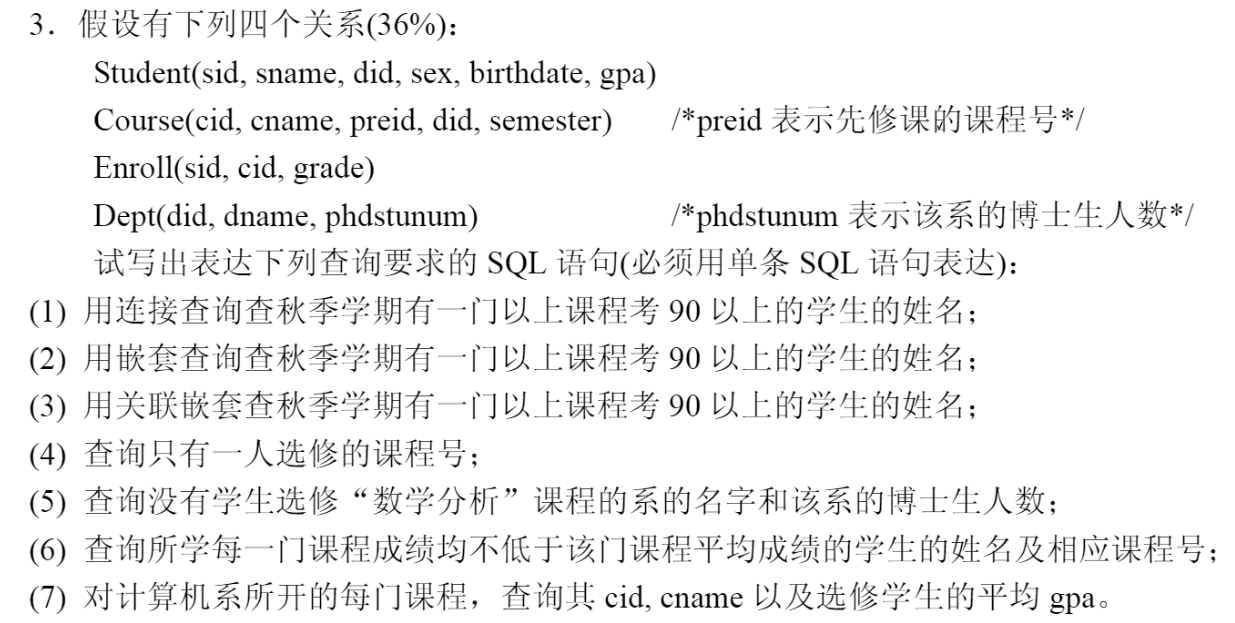


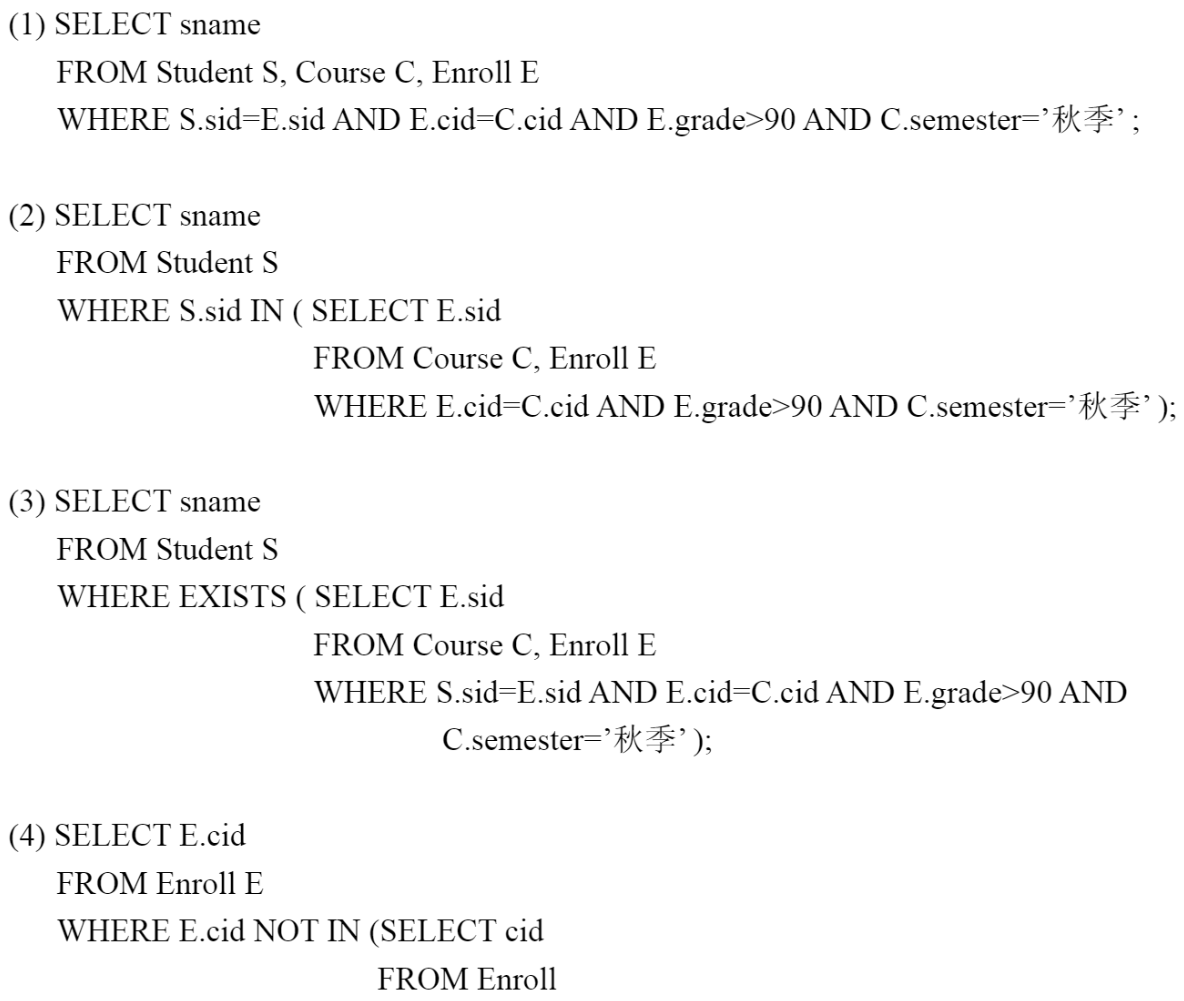




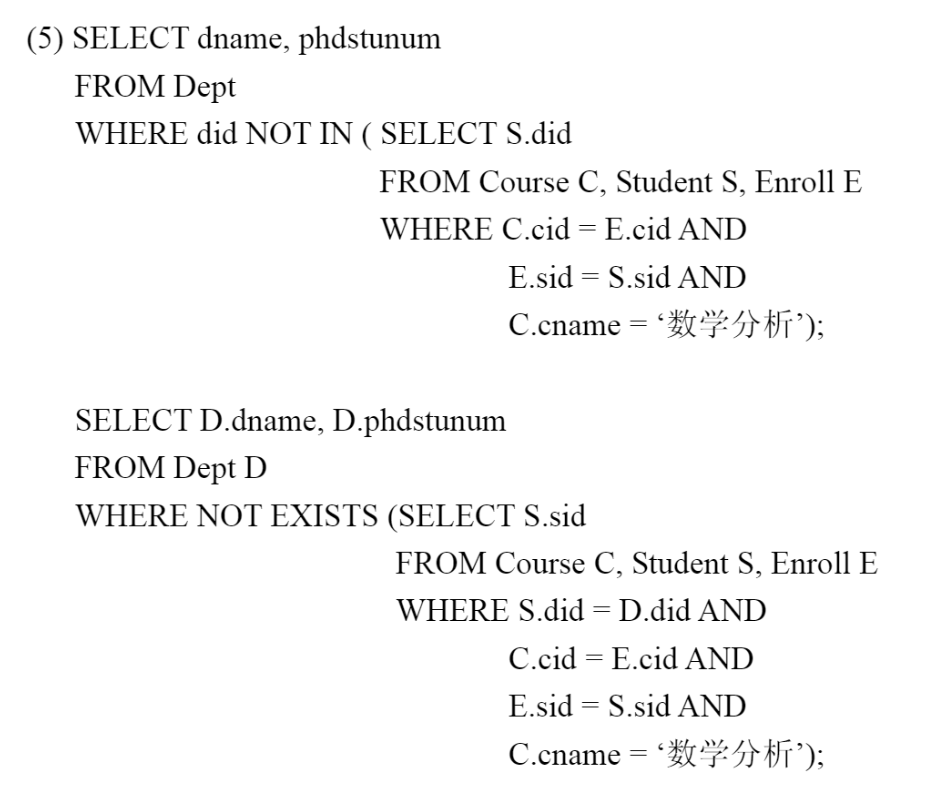


考试3



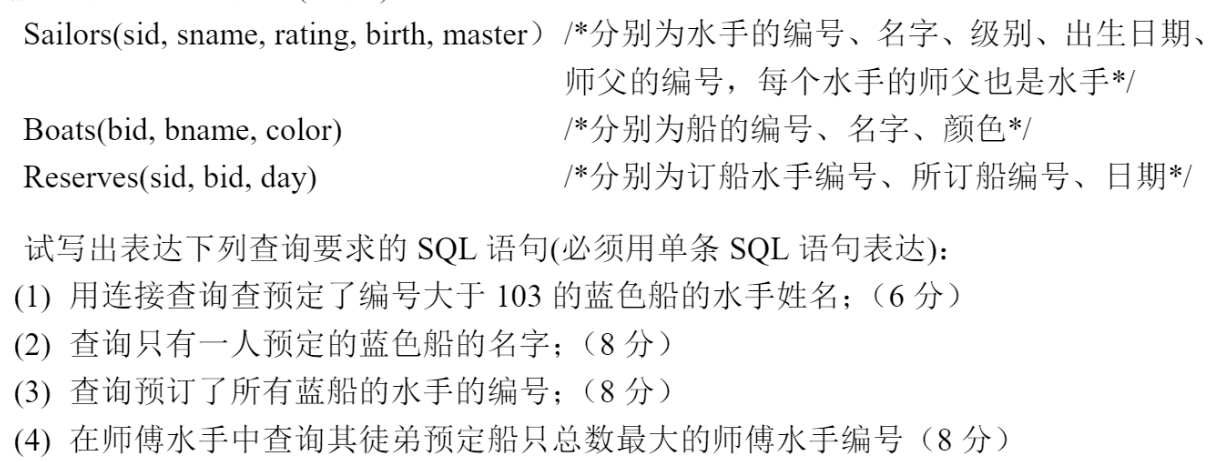




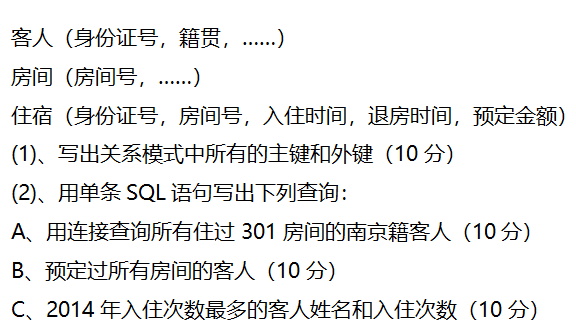




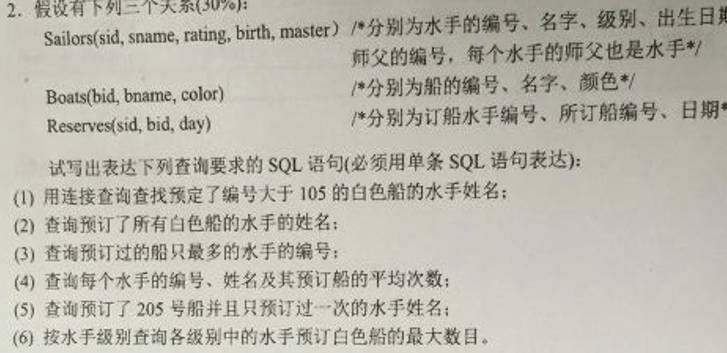
考试4



考试5



考试6



(1)

SELECT S.sname FROM Sailors S, Boats B, Reserves R

WHERE S.sid=R.sid AND B.bid=R.bid AND R.bid>105 AND B.color=’white’

(2)

SELECT S.sname FROM Sailors S

WHERE NOT EXISTS

(SELECT B.bid FROM Boats B WHERE NOT EXISTS

(SELECT R.bid FROM Reserves R WHERE R.bid=B.bid AND R.sid=S.sid))

(3)

SELECT TEMPR.sid FROM

(SELECT R.sid, COUNT(R.bid) AS count

FROM Reserves R GROUP BY R.sid) AS TEMPR

WHERE TEMPR.count=(SELECT MAX(TEMPR.count) FROM TEMPR)

SELECT R.sid FROM Reserves R

GROUP BY R.sid HAVING COUNT(\*)=

(SELECT MAX(count) FROM

(SELECT COUNT(\*) AS count FROM Reserves R1 GROUP BY R1.sid))

(4) (平均次数是什么鬼)

SELECT R.sid, S.sname, avgcount=COUNT(R.bid)/(MAX(day)-MIN(day))

FROM Reserves R, Sailors S

WHERE S.sid=R.sid

GROUP BY R.sid

(5)

SELECT S.sname FROM Reserves R, Sailors S

WHERE S.sid=R.sid

GROUP BY R.bid, R.sid, S.sname

HAVING R.bid=205 AND COUNT(\*)=1

(6)

SELECT S.rating, COUNT(R.bid) AS cnt

FROM Reserves R, Sailors S

WHERE S.sid=R.sid AND R.color='white'

GROUP BY S.rating, R.sid

HAVING COUNT(R.bid) >= ALL(

SELECT COUNT(R1.bid)

FROM Reserves R1, Sailors S1

WHERE S1.sid=R1.sid AND R1.color='white'

GROUP BY S1.rating, R1.sid

HAVING S1.rating=S.rating)

考试6



(1)

SELECT S.sid, S.sname FROM student S, enroll E

WHERE E.dname1=’Computer Sci’ AND E.grade>3 AND S.sid=E.sid

(2)

SELECT \* FROM course C WHERE C.cno IN

(SELECT E.cno FROM enroll E GROUP BY E.cno HAVING COUNT(\*)=1)

(3)

SELECT AVG(E.grade) FROM enroll E

GROUP BY E.cno, E.dname1 HAVING E.dname1=’Computer Sci’

(4)

SELECT S.sname, S.sid FROM student S

WHERE NOT EXISTS

(SELECT C.cno FROM course C WHERE C.dname=’Computer Sci’

AND NOT EXISTS

(SELECT E.cno FROM enroll E WHERE E.sid=S.sid AND E.cno=C.cno))

(5)

SELECT E.grade, E.sid FROM enroll E,

(SELECT MAX(E1.grade) AS maxgrade, E1.cno FROM enroll E1

WHERE E1.dname1=’Computer Sci’ GROUP BY E1.cno) AS TEMP

WHERE E.dname1=’Computer Sci’

AND E.cno=TEMP.cno AND E.grade=TEMP.maxgrade

SELECT E.grade, E.sid FROM enroll E

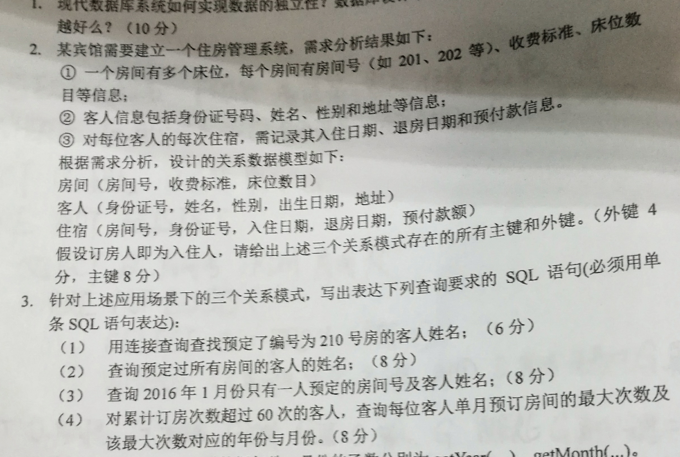
WHERE dname1=’Computer Sci’ AND E.grade=

(SELECT MAX(E1.grade) FROM enroll E1

WHERE dname1=’Computer Sci’

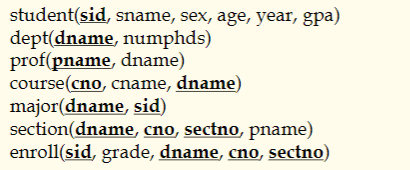
GROUP BY E1.cno HAVING E1.cno=E.cno)

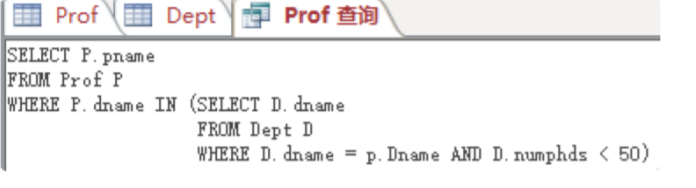
考试7



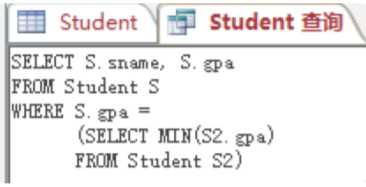
（4）select month, year, id, cnt\_month  
  
from   
  
(select count(\*) cnt\_month, id, MONTH(date) month, YEAR(date) year  
  
from 住宿  
  
group by id, MONTH(date), YEAR(date)  
  
where id in  
  
(select id  
  
from 住宿  
  
group by id  
  
having count(\*)>60)) tempTable  
  
where (cnt\_month, id) in  
  
(select max(cnt\_month), id  
  
from tempTable  
  
group by id)

实验

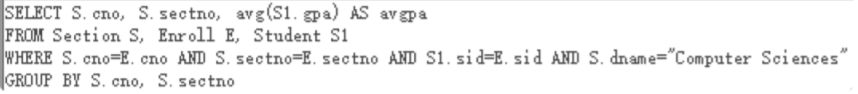


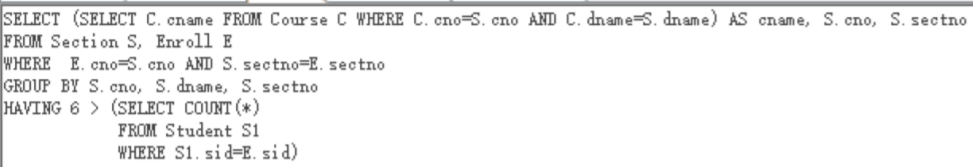
1.打印在博士学生少于50人的系工作的教授的姓名。

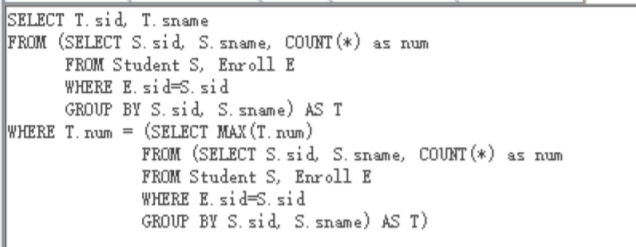
2.打印gpa最低的学生的姓名。



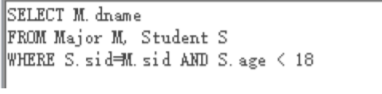
3.

对于每个计算机科学类，打印cno、sectno和的平均gpa

4.打印所有注册学生少于6人的班级的课程名称、课程编号和区号。

5.打印参加大多数课程的学生的姓名和签名。

6.打印出有一个或多个专业且年龄在18岁以下的部门的名称。



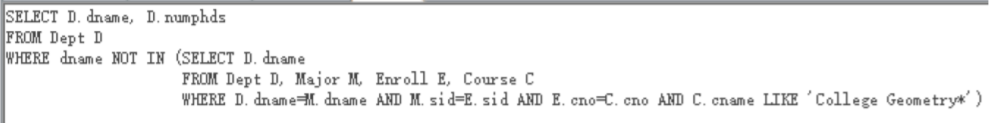
1. Print the names and majors of students who are taking one of the College Geometry courses. (Hint: You'll need to use the "like" predicate and the string matching character in your query.)

打印正在学习大学几何课程的学生的名字和专业。(提示：您需要在查询中使用“like”谓词和字符串匹配字符。)



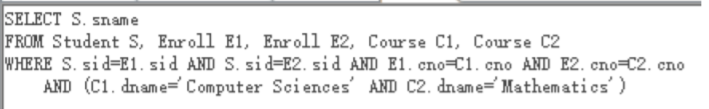
1. For those departments that have no majors taking a College Geometry course, print the department name and the number of PhD students in the department.

对于那些没有专业参加大学几何课程的系，打印系名和系内的博士生人数。



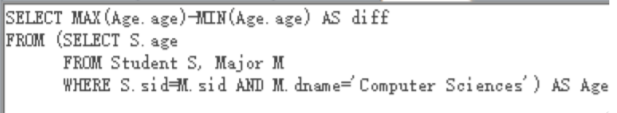
1. Print the names of students who are taking both a Computer Sciences course and a Mathematics course.

打印出同时参加计算机科学课程和数学课程的学生的名字。



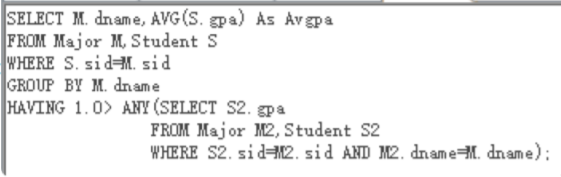
1. Print the age difference between the oldest and youngest Computer Sciences major(s)

打印最年长和最年轻的计算机科学专业之间的年龄差异（年代）



1. For each department that has one or more majors with a GPA under 1.0, print the name of the department and the average GPA of its majors.

对于每个有一个或多个专业的平均绩点低于1.0的部门，打印出该部门的名称和其专业的平均绩点。



1. Print the ids, names, and GPAs of the students who are currently taking all of the Civil Engineering courses

打印目前正在学习所有土木工程课程的学生的id、姓名和gpa

