CS 186 , Spring 1996
Midterm#1 (2/8/96)

There are 7 questions to this exam. All count equally. Each question is worth 6 points.

Queries expressed in relational algebra and relational calculus must follow the syntax used in class. Queries expressed in SQL must follow the syntaxused in class or in the Sybase Manual. All SQL queriesshould contain NO duplicates in their outputs. It is not necessary to short the outputs in any order.

For each of the mutiple choice questions, you should choose ONLY ONE answer. If you choose more than one answer or the wrong answer, you will lose more points than if you do not answer the question. For example, fore question(1), you will receive all 6 pointes if you choose the correct answer, 0 points if you choose no answer, and -2 points if you choose mutiple answers or the wrong answer.

Problem #1

Consider the database schema used in HW #2:

DEPT(dname,location)
EMP
PROJECT(pname,budget,manager)
JOBS(emp_no,pname)

This database stores the current research projects at UC Berkeley. People from different departments works together on projects. Each project has a unique name and one manager. It is possible that an employee is working on or managing multiple projects. We assume that every employee has a unique emp_no. Employee names are also unique. The manager field of PROJECT stores the emp_no(not the name) of project managers.
There are three options in enforcing referential integrity: reject, cascade, and nullify. Suppose an employee is deleted from the EMP relation, circle the statment below which is correct.

a. If reject is chosen, the delete should be rejected if the employee manages one or more projects.
b. If reject is chosen, the delete should be rejected if the employee's department has other employees.
c. If cascade is chosen, the projects that the employee works on should also be deleted, so are the managers of these projects.
d. If nullify is chosen, the employee's dname attribute should be set to null.
e. None of above.

Problem #2

Consider the following relational database schema which is used for examples in class:

DEPT(dname,location)
STUDENT(name,regno,gpa,level,dept)
COURSE(cno,cname,dept)

TAKE(regno,cno)

The following relational algebra queries are supposed to return the names of the students from the Math department who are taking CS course. Please circle the correct answer.

a. (STUDENT where dept = "Math")[name, regno] join TAKE join ((COURSE where dept = "CS")[cno])
b. ((STDUENT where dept = "Math") join TAKE join (COURSE where dept = "CS"))[name]
c. ((STUDENT where dept = "Math")[name, regno] join TAKE join ((COURSE where dept = "CS")[cno]))[name]
d. ((STUDENT where dept = "Math")[name, regno, dept] join TAKE join ((COURSE where dept = "CS")[cno, dept]))[name]
e. None of above

Problem #3

Consider the following relational database schema which is the same as in(2):

DEPT(dname,location)
STUDENT(name,regno,gpa,level,dept)
COURSE(cno,cname,dept)
TAKE(regno,cno)

Assume GPA values are unique, express in relational algebra the regno of the student who has the highest GPA.

Problem #4

Consider the following relational database schema which is the same as in(2):

DEPT(dname,location)
STUDENT(name,regno,gpa,level,dept)
COURSE(cno,cname,dept)
TAKE(regno,cno)

Assume GPA values are unique, express in relational calculus the regno of the student who has the second highest GPA. Circle the correct answer.

a. s.regno where exists s1 (s1.sid s.sid and s1.gpa > s.gpa) and not exists s2 (s2.sid s.sid and s2.gpa > s.gpa)
b. s.regno where exists s1 (s1.gpa > s.gpa and not exists s2 (s2.sid s1.sid and s2.gpa > s.gpa))
c. s.regno where exists s1 ((forall s2 (s1.gpa > s2.gpa)) and s.gpa < s1.gpa)
d. s.regno where exists s1 (forall s2 (s1.gpa > s2.gpa)) and not exists s3 (s3.gpa > s.gpa)
e. None of above

Problem #5

Cnsider the following relational database schema which is the same as in (2):

DEPT(dname,location)
STUDENT(name,regno,gpa,level,dept)
COURSE(cno,cname,dept)

TAKE(regno,cno)

Express in SQL the names of the students who take exactly the same courses as Doug takes.

Problem #6

Consider the following relational database schema which is the same as in(2):

DEPT(dname,location)
STUDENT(name,regno,gpa,level,dept)
COURSE(cno,cname,dept)
TAKE(regno,cno)

Express in SQL the names and departments of the students who take the largest number of courses among students in their departments. Students without a department should not be considered.

Problem #7

Consider the following relational database schema which is the same as in(2):

DEPT(dname,location)
STUDENT(name, regno, gpa, level, dept)
COURSE(cno,cname,dept)
TAKE(regno,cno)

Rewrite the following SQL query in relational algebra:

```
select distinct s.name
from STUDENT s
where s.level = 4 and
      not exists
      (select c.*
       from COURSE c
       where c.dept = 'CS' and
             not exists (
                   select t.*
                   from TAKE t
                   where t.regno = s.regno
                          and t.cno = c.cno)
```

Circle the correct answer:

a. ((STUDENT where level 4) > join TAKE join (COURSE where dept "CS"))[name]
b. (((STUDENT where level = 4) join TAKE)[name,cno]) divideby (COURSE where dept = "CS")[cno]
c. ((STUDENT where level = 4) minus (STUDENT join TAKE join COURSE where dept "CS"))[name]
d. (((STUDENT where level = 4) join TAKE) divideby (COURSE where dept = "CS")[cno])[name]
e. None of above