CS 186, Spring 1995
Midterm
Professor Wang

There are a total of 8 questions. Credits for each question are indicated in brackets. There are a total of 45 points.

Queries expressed in relational algebra and relational calculus must follow the syntax used in class. Queries expressed in SQL must follow the syntax used in class or in the INGRES Manual. All SQL queries should contain NO duplicated in their outputs. It is not necessary to sort the outputs in any order.

You are NOT allowed to use views for any questions

Problem #1

(1) [5] Given a relation with 3 attributes, at most how many candidate keys can it have at any given time? At least how many candidate keys does it have? Please state your reasoning in support of your answer.

Problem #2

(2) Consider the following relational database schema:

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

None of the columns can have null values.

Problem #2a

(2a)[5] Express in relational algebra the names of the departments which have at least one student who does not take any courses offered by the 'EECS' department.

Problem #2b

(2b) [5] Express in relational calculus the names of the courses such that everyone taking them is an EECS senior.

Problem #3

(3) [5] Consider the following relational database schema:

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

None of the columns can have null values.
Express in SQL the names of the departments which have at least one student who does not take any courses offered by the 'EECS' department.

Problem #4

(4) [5] Consider the following relational database schema:

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

For each relation, the attributes in *italics* are the primary key. None of the columns can have null values. Express in SQL the names of the projects which have the largest number of employees from the EECS department.

Problem #5

(5) [5] Consider the following relational database schema:

```
DEPT (dname, location)
STUDENT (name, regno, gpa, level, dept)
```

For each relation, the attributes in *italics* are the primary key. Only the gpa and dept attributes in STUDENT can have null values. You may assume the non-null gpa values are unique.

Express in SQL the name of the students who have the 4$^{\text{th}}$ highest gpa in their deparments.

Problem #6

(6) [5] Consider the following relational database schema:

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

For each relation, the attributes in *italics* are the primary key. Only the gpa and dept attributes in STUDENT can have null values.
Express in SQL the names of the departments such that every senior from these departments is taking all the EECS courses offered.

Problem #7

(7) [5] Consider the following relational database schema:

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

For each relation, the attributes in *italics* are the primary key. Only the gpa and dept attributes in STUDENT can have null values. You may assume that the DEPT, STUDENT, and COURSE tables are not changed. Updates, inserts, and deletes are allowed on the TAKE table.
Write the rules and the associated stored procedures to enforce the following integrity constraints: - If a student has a gpa lower than 2.0, s/he is not allowed to take more than 12 units. - If a student does not have a gpa, s/he is not allowed to take more than 15 units.

Problem #8

(8) [5] What are the different types of integrity constrains? Which of them cannot be implemented by using INGRES rules and stored in procedures? Please state your reasoning in support of your conclusions.