CS4604 Final Exam

May 10, 2000

Please enter the following information:

- **Name:**

- **ID:**

GOOD LUCK and have a nice summer!!
Do not write below this line

Problem	Max Score	Score
1	30	
2	10	
3	10	
4	20	
5	10	
6	10	
7	10	
8	15 (XC)	
Total	100	

1. (30 points) Short answer questions:

   (a) What property of decompositions is guaranteed by both BCNF and 3NF?

   (b) Carefully state the rule to determine if a given Datalog query is safe.

   (c) Assume that the FD $AB \rightarrow C$ holds in a relation $R(A, B, C)$. Express this FD as a constraint in relational algebra.

   (d) Give two examples of active database elements.

   (e) Why do most RDBMS implementations provide bag-semantics as default?

   (f) Consider a relation $R(A_1, A_2, A_3, A_4, \ldots, A_n)$. What is the maximum number of keys R can have at the same time?

   (g) Why are there no 'weak' classes in ODL?

   (h) Complete the following sentence: $\{A_1, A_2, \ldots, A_n\}^+$ is the set of all attributes of R if and only if $A_1, A_2, \ldots, A_n$ is a _____ for R.

   (i) Examples were given in class of *useless*, *obvious*, or *uninteresting* patterns discovered by data mining. Mention any one of them.

   (j) All the letters of the word *database* are typed with the left hand on the QWERTY keyboard. What is the significance of this aspect? :-)

2. (10 points) Design an E/R diagram for the following situation: Doctors prescribe drugs for patients. A given doctor can prescribe many drugs for a certain patient. A patient might be treated by many doctors and can be prescribed the same drug by different doctors. Drugs can only be purchased by presenting prescriptions to the pharmacist. A prescription can involve more than one patient (e.g. a mother and her baby), more than one drug, but is associated with a unique doctor.

3. (10 points) Consider the relation $R(A, B, C, D, E)$ with FDs $\{A \rightarrow BC, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$. Now consider two decompositions of R:

   - $\{ABC\}, \{ADE\}$
   - $\{ABC\}, \{CDE\}$

   It is easy to verify that the first decomposition provides a lossless-join while the second decomposition is not lossless. Using these two examples, give a general rule to determine if a given decomposition is a lossless-join decomposition.

   Notice that we are not looking for a way to perform a lossless-join decomposition (we can simply use the 3NF or BCNF rules); rather we are given a decomposition and we would like to verify if it is a lossless-join.

4. (a) (8 points) Consider the relation `Advisor(student-name,advisor-name)`. Each student has only one advisor but one advisor can advise many students. Write a query in Datalog to find those students whose advisors do not advise any other students.

(b) (6 points) Suppose R and S have n and m tuples respectively. Give the minimum and maximum numbers of tuples that $R \bowtie S$ can have.

(c) (6 points) Under what conditions will $(R \cup S) - T \equiv (R - T) \cup (S - T)$ not hold?

5. (10 points) An operator on relations is called *monotone* if whenever we add a tuple to one of its arguments, the result contains all the tuples that it contained before adding the tuple, plus perhaps more tuples. Which of the following operators are monotone? For each, either explain why it is monotone or give an example showing it is not.

   - $\times$

   - $\bowtie$

   - $\cap$

   - $-$

   - σ

6. (10 points) Consider the relational schema:

   ```
   HasTaken(student-ssn,course-number)
   RequiredForGraduation(course-number)
   ```

   with their obvious interpretations *i.e.,* `HasTaken` identifies the courses taken by each student and `RequiredForGraduation` lists all the courses required to graduate. For example, if `RequiredForGraduation` contains 14 courses, this means that all those 14 courses are required to graduate (you may assume that all the `course-number`s in `RequiredForGraduation` will be different). Write a query in relational algebra to find the students who have satisfied **all but one** of the requirements for graduation.

7. (10 points) Consider the relation `Student(id,name,age,gpa)`. Write a query in SQL to find the names of students for whom there are at least two more older students with the same `gpa`.

8. (Extra credit question: 15 points) Assume that you are given a relation `Grades(id,gpa)` that lists a student's `id`, followed by his/her `gpa`. Write an SQL query to find the `gpa` that occurs most often in the `Grades` relation. If two `gpa`s occur equally often, your query should output both of them.