Stanford University
Computer Science Department

Fall 2002 Comprehensive Exam in Databases

1. **OPEN Book & Notes:** no laptops, Internet access, etc.
2. **Write only on the answer sheet provided.**
3. **The exam is timed for one hour.**
4. **Write your Magic Number on this sheet & on the Exam paper.**

The following is a statement of the Stanford University Honor Code:

A. The Honor Code is an undertaking of the students, individually and collectively:

1. that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;

2. that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.

B. The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.

C. While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.

By writing my Magic Number below, I certify that I acknowledge and accept the Honor Code.

Magic Number----------------------------------

Stanford University Computer Science Department
2002 Comprehensive Exam in Databases

- The exam is *open book and notes.*
- There are 7 problems on the exam, with a varying number of points for each problem and subproblem for a total of 60 points (i.e., one point per minute). It is suggested that you look through the entire exam before getting started, in order to plan your strategy.
- Please write your solutions in the spaces provided on the exam. Make sure your solutions are neat and clearly marked.
- *Simplicity and clarity of solutions will count.* You may get as few as 0 points for a problem if your solution is far more complicated than necessary, or if we cannot understand your solution.

MAGIC NUMBER: ____________

Problem	1	2	3	4	5	6	7	TOTAL
Max. points	6	12	6	10	6	8	12	60
Points								

1. **(6 points)** Consider an Entity-Relationship diagram with a 3-way relationship R connecting entity sets A, B, and C. In the E/R diagram, there are arrows entering B and C from R, but no arrow entering A. Suppose the number of entities currently in entity sets A, B, and C are a, b, and c, respectively. Also suppose that there are currently r triples in the relationship set for R.

   (a) It must be that $r \leq ab$. Explain why.

   (b) What other nontrivial constraints on values a, b, c, and r must hold?

2. **(12 points)** Suppose we have a relation $R(A, B, C, D, E, F, G, H, I, J)$ with the following nine functional dependencies:

$$\begin{aligned} FI &\rightarrow C \\ AC &\rightarrow I \\ EJ &\rightarrow H \\ GI &\rightarrow E \\ CJ &\rightarrow E \\ DE &\rightarrow B \\ EF &\rightarrow H \\ BE &\rightarrow A \\ IJ &\rightarrow D \end{aligned}$$

   (a) Compute the closure $(CFGJ)^+$, that is, the set of attributes that must be equal in any two tuples that agree on all of C, F, G, and J.

(Problem continues on next page)

(b) Three of the ten attributes must be in any key. Which three are they, and why? (There is a brief, simple, explanation that does not require you to find all the keys.)

(c) Find the smallest key, and explain why there is no other key of *the same size* or smaller.

3. **(6 points)** Suppose relation $R(A, B, C, D, E)$ satisfies the multivalued dependencies $A \rightarrow\!\!\!\rightarrow B$ and $B \rightarrow\!\!\!\rightarrow D$. Also suppose that R contains the tuples $(0, 1, 2, 3, 4)$ and $(0, 5, 6, 7, 8)$. List the tuples in the smallest possible relation that R could be.

4. **(10 points)** In the following, all outerjoins should be assumed "natural."

   (a) If we use the set model of relations, then it is possible to express the full outerjoin of two relations as the union of the left-outerjoin and right-outerjoin of these two relations. Explain why this formula produces the full outerjoin.

   (b) Show that the the formula from (a) does not work if we use the bag (multiset) model of relations.

   (c) For the bag model, is there any way to express the full outerjoin in terms of the left- and right-outerjoin and other operations of relational algebra? Either give such a formula or show that none exists.

5. **(6 points)** Consider relations $R(A, B, C)$ and $S(A, B, C)$ with no assumptions about keys. For the following two statements using relational algebra, state the constraint or property that is specified by the given statement. Please be specific about the actual attributes involved. In both cases the correct answer corresponds to a specific database concept and can be stated in a few words. Much longer answers will be considered incorrect.

   Note: Symbol $\rhd\!\!<$ is the *semijoin* operator. $E_1 \rhd\!\!< E_2 \equiv \pi_{schema(E_1)}(E_1 \bowtie E_2)$.

   (a) $\pi_A(R) \rhd\!\!< \pi_A(S) = \pi_A(R)$

   (b) $\sigma_{B1 \neq B2}(\rho_{R1(A,B1,C1)}(R) \bowtie \rho_{R2(A,B2,C2)}(R)) = \emptyset$

6. **(8 points)** Consider the following relations in a SQL database:

```
PhDstud(name,year,email)    // name is a key
Aligned(student,adviser)    // student is a foreign key
                               referencing PhDstud.name
```

   Suppose the owner (creator) of these relations is a user named "Kathi," and Kathi wants to grant to a user named "Rajeev" the ability to read all information about all first-year PhD students who are not aligned with an adviser (and only those students). Specify a command or sequence of commands that achieves this goal.

7. **(12 points)** Consider a relation `Visit(inspector,restaurant,date)` recording visits inspectors make to certain restaurants on certain dates. Assume no more than one inspector visits any given restaurant on a given date. Values for the `date` attribute can be compared using =, <, <=, etc., and aggregate functions `min`, `max`, and `count` may be applied to the `date` column. Write a SQL query that returns the two most recent visit dates to each restaurant.

   The result of your query should be an ordered two-column relation, where each restaurant appearing in the original `Visit` relation has two adjacent tuples, with the more recent visit for each restaurant listed first. If a restaurant has only one visit, your result should still include a single tuple for that restaurant. For example, we might see the following result, where "Thai Cafe" has only one visit:

restaurant	date
Bytes	4/1/02
Bytes	3/31/02
Coffee House	9/15/01
Coffee House	8/1/00
Thai Cafe	1/10/02
Tresidder	3/10/01
Tresidder	3/9/01

   Your SQL query will be graded on simplicity as well as correctness.