Stanford University
Computer Science Department

Fall 2005 Comprehensive Exam in Databases

1. **Open Book and Notes - NO laptop. Write your solutions in the spaces provided on the exam.**
2. **The exam is timed for one hour.**
3. **Write your Magic Number in the space provided on this sheet and the following sheet; DO NOT WRITE YOUR NAME.**

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 un-permitted 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
2005 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	12	8	6	8	10	8	8	60
Points								

1. *Relational Algebra* **(12 points)**. In this question, we use $t[X]$ to represent the components of tuple t in attributes X. Let relation R have schema XY (i.e., the set of attributes $X \cup Y$) and let relation S have schema YZ. Assume X, Y, and Z are disjoint. The *strong join* of R by S is the set of tuples $t[X]$ such that t is a tuple of R, and for all tuples r in R (including t) such that $r[X] = t[X]$, there is some tuple s in S such that $r[Y] = s[Y]$. Write, as a sequence of assignments of relational-algebra expressions to variables, a program that computes the strong join of $R(A, B)$ by $S(B, C)$. **Note**: you should use only the basic relational-algebra operators: select, project, union, intersection, difference, product, natural join, and theta-join. Also, please explain what each step is doing, if you want to be considered for partial credit.

2. *Functional Dependences* **(8 points)**. Armstrong's axioms for functional dependencies are:

   (a) *Reflexivity*: If $X \subseteq Y$, then $Y \rightarrow X$.

   (b) *Augmentation*: If $X \rightarrow Y$, then $XZ \rightarrow YZ$.

   (c) *Transitivity*: If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$.

   From these axioms, prove the *union rule*: If $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow YZ$. Note that X, Y, and Z are sets of attributes, XY denotes $X \cup Y$, and nothing about disjointness is assumed.

3. *ODL and E/R Models* **(6 points)**. Your answers to (a) and (b) will be judged not only on being truthful, but also on selecting among all options the things that are most significant.

   (a) **(4 points)** Name two things that object-oriented models such as ODL offer the designer and that are not found in the E/R model.

   (b) **(2 points)** Name one thing that the E/R model offers that ODL does not.

4. *XML* **(8 points)**. Here is a DTD:

```
<!DOCTYPE a [
    <!ELEMENT a (b+, c*)>
    <!ELEMENT b (c?)>
    <!ELEMENT c (#PCDATA)>
]>
```

   (a) **(3 points)** Give a shortest (fewest number of elements) example of a sequence of opening and closing tags that is valid for this DTD.

   (b) **(5 points)** Give a shortest (fewest number of elements) example of a well-formed sequence of opening and closing tags where a is the root tag, b and c tags are also present, but the sequence is *not* valid for this DTD.

5. *Multivalued Dependencies and SQL Constraints* **(10 points)**. Consider a table $R(A, B, C)$. Make no assumptions about keys.

(a) **(6 points)** Using SQL Assertions, write an assertion stating that the multivalued dependency $A \twoheadrightarrow B$ holds on R.

(b) **(4 points)** Can you enforce the same dependency using SQL Tuple-Based `CHECK` constraints? If so, show the constraint(s). If not, explain why not.

6. *SQL Equivalence* **(8 points)**. Consider the following two SQL queries over tables $R(A, B)$ and $S(C)$. Make no assumptions about keys.

```
Q1: select A from R                    Q2: select distinct A from R, S
    where B in (select C from S)           where R.B = S.C
```

Are these two queries equivalent? That is, do they return the same answer on all possible instances of R and S? If so, justify why. If not, show a counterexample.

7. *Transactions* **(8 points)**. Consider a relation `Emp(ID,salary)` storing employee IDs and salaries, where ID is a key. Consider the following two transactions:

```
T1: begin transaction
    update Emp set salary = 2*salary where ID = 25
    update Emp set salary = 3*salary where ID = 25
    commit

T2: begin transaction
    update Emp set salary = 100 where salary > 100
    commit
```

Suppose the salary of the employee with ID=25 is 100 before either transaction executes.

(a) **(4 points)** If both transactions T1 and T2 execute to completion with isolation level *serializable*, what are the possible final salaries for the employee with ID=25?

(b) **(4 points)** Now suppose transaction T1 executes with isolation level *read-committed*, transaction T2 executes with isolation level *read-uncommitted*, and both transactions execute to completion. What are the possible final salaries for the employee with ID=25?