Computer Science Department
Stanford University
Comprehensive Examination in Databases
Autumn 1994

October 21, 1994

READ THIS FIRST!

You should write your answers for this part of the Comprehensive Examination in a BLUE BOOK. Please use a separate blue book for each problem. Be sure to write your MAGIC NUMBER on the cover of every blue book that you use.

The total number of points is 30, and the exam takes 30 minutes. This "coincidence" can help you plan your time.

This exam is OPEN BOOK. You may use notes, articles, or books—but no help from other sentient agents such as other humans or robots.

Show your work, since PARTIAL CREDIT will be given for incomplete answers. For example, you can get credit for making a reasonable start on a problem even if the idea doesn't work out; you can also get credit for realizing that certain approaches are incorrect. On a true/false question, you might get partial credit for explaining why you think something is true when it is actually false. But no partial credit can be given if you write nothing.

1994 Comprehensive Exam in Databases
Three problems, 30 points total

1. **(15 points)** Consider the following self-explanatory relational database schema:

```
student(ID, name, address)     /* ID and <name,address> are both keys */
course(dept, code, units)      /* <dept,code> is a key */
enrolled(ID, dept, code)       /* <ID,dept,code> is a key */
```

   (a) Write a relational algebra expression that returns the name and address of each student enrolled in at least one course in the "CS" department.

   (b) Write an SQL query that returns the name and address of each student enrolled in at least one course in the "CS" department.

   (c) Will (a) and (b) return exactly the same result? Briefly explain your answer.

   (d) Write an SQL query that returns the total number of units enrolled in by all of the students in the database.

   (e) Can the query for part (d) be written in relational algebra? Briefly explain your answer.

2. **(5 points)** Suppose we have five attributes, A,B,C,D,E, with the following functional dependencies:

$$\begin{aligned} A &\rightarrow D \\ AB &\rightarrow C \\ B &\rightarrow E \\ D &\rightarrow C \\ E &\rightarrow A \end{aligned}$$

   Consider the relation schema ABC. What are all of the keys for ABC? Justify your answer.

   Hint: Be sure to consider all five attributes when determining the functional dependencies for ABC.

3. **(10 points)** Transactions often read database items before they know if they need to modify them. For example, a transaction might first read the number of available seats on a flight, and if it is positive, decrease this count to reserve a seat. This pattern leads to the common deadlock illustrated by this example:

```
Transaction T1 requests read lock for item A -> request granted
Transaction T2 requests read lock for item A -> request granted
Transaction T1 requests write lock for A -> T1 must wait
Transaction T2 requests write lock for A -> T2 must wait -> deadlock
```

   To avoid these deadlocks, some systems implement an *update* lock. This lock gives read privileges over an item, just like a read lock. However, by requesting an update lock, a transaction declares that it is likely that it will request a write lock on this item later on.

   (a) Give a lock compatibility matrix for read, write, and update locks.

   (b) Using your compatibility matrix from part (a) and the example above, show how the deadlock is avoided. Be sure to show when each transaction gets its write lock.