CS145 Midterm Examination

Monday, November 2, 1998, 11AM–12:15PM

Directions

The exam is *open book*; any written materials may be used. Answer all six questions on the exam paper itself. The total number of points is 100. When SQL is called for, you *must* use standard SQL2, not the Oracle version. Do not forget to **sign the pledge** below.

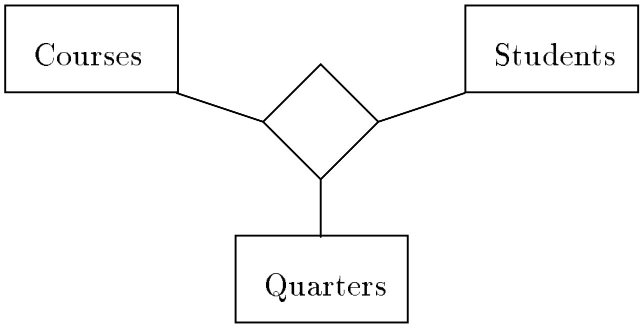
I acknowledge and accept the honor code.

____

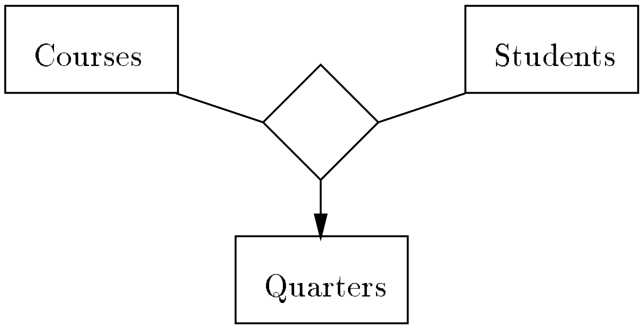
Print your name here: ____

1	2	3	4
5	6		

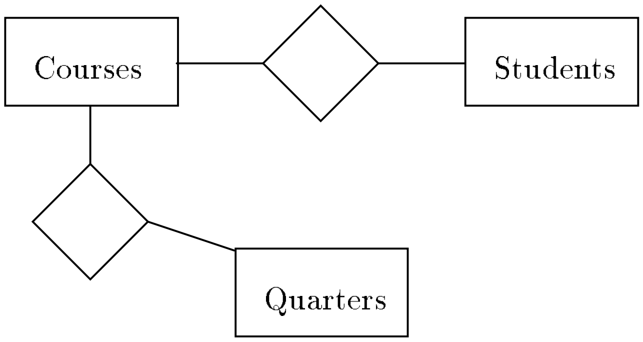
Problem 1: (15 points) In this problem, we assume you are all familiar with courses, students, and quarters. We have drawn several E/R diagrams intended to describe a database telling which students take which courses in which quarters. We ask you to discuss the differences among different diagrams in practical terms, e.g., "a student can only take one course at a time," rather than in abstract terms such as "there is a many-one relationship from students to courses."



I



II



III

a) Explain the important difference between the diagrams I and II.

b) Suppose we were to modify diagram II by adding another arrow, this one on the line to `Courses`. Describe the constraint that would be implied by the new diagram.

c) Give two limitations of diagram III that are not present in diagram I.

Problem 2: (15 points) Give an ODL design that describes the following kinds of objects. There are "people," with a name and address. Some people are single, others are married. Married people have a spouse, which the database must indicate. Single people are either never-previously-married, widowed, or divorced. For those who have been previously married but are now single, we want the database to indicate all their previous spouses. *Do not include any useless subclasses in your design.* You may use the following abbreviations: att = attribute; rel = relation; intf = interface.

Problem 3: (15 points) Suppose we are given the set of functional dependencies $A \to B$, $B \to C$, $C \to D$, $D \to E$, and $E \to A$.

a) If we project these FD's onto the set of attributes ABC, what are *all* the nontrivial FD's for that relation? You need not report trivial FD's or FD's with more than one attribute on the right, but you should report all other FD's.

b) What are all the keys for ABC? Do not report superkeys that are not (minimal) keys.

c) Of the 10 subsets of three out of the five attributes $ABCDE$, how many are in BCNF with respect to the given FD's? ________

Problem 4: (15 points) Suppose we have a relation

`Computers(ipAddr, name, manf, model, location)`

Write the following queries *in relational algebra*. You may use temporary relations to describe your queries.

a) Find the IP address of the computer named "Shalmaneser."

b) Find all pairs of computers that have the same location. You should list each pair only once.

c) Find all the manufacturers for which there are computers with at least three different model numbers.

Problem 5: (10 points) For an SQL implementation, we have decided to split the relation of Problem 4 into two relations:

```
Comp1(ipAddr, name, location)
Comp2(ipAddr, manf, model)
```

Write SQL2 declarations for `Comp1` and `Comp2` that express the following constraints:

a) Both `ipAddr` and `name` uniquely specify an individual computer.
b) If an IP address appears in `Comp2`, then it must also appear in `Comp1`.
c) The location may not be null.
d) The manufacturer is either SUN, HP, or Apple.

Pick appropriate types for attributes.

```
CREATE TABLE Comp1 (
```

```
);
```

```
CREATE TABLE Comp2 (
```

```
);
```

Problem 6: (30 points) Using the two relations from Problem 5, write the following *standard SQL2* statements:

a) Find the manufacturer of the computer named Shalmaneser.

b) List all the computer names and IP addresses in alphabetical order of their names.

c) For each manufacturer, find the number of different models in the database.

d) For each manufacturer, find the number of different *computers* in the database, but only do so if the manufacturer has at least two different models.

e) Create a view of the data giving for each computer name the manufacturer and location.

f) Constrain the relations so that there cannot be more than 10 computers by the same manufacturer at the same location.