CS145 Midterm Examination

Monday, November 6, 2000, 11AM–12:15PM

Directions

- The exam is *open book*; any written materials may be used.
- Answer all 3 questions on the exam paper itself.
- The total number of points is 75 (i.e., one point per minute of work).
- When SQL is called for, you *must* use standard SQL2, not the Oracle version.
- You will be graded not only on correctness of queries and other database code, but on the simplicity of your answers. You may get as little as 0 points for an answer that "works" but is excessively complicated.
- Do not forget to **sign the pledge** below.

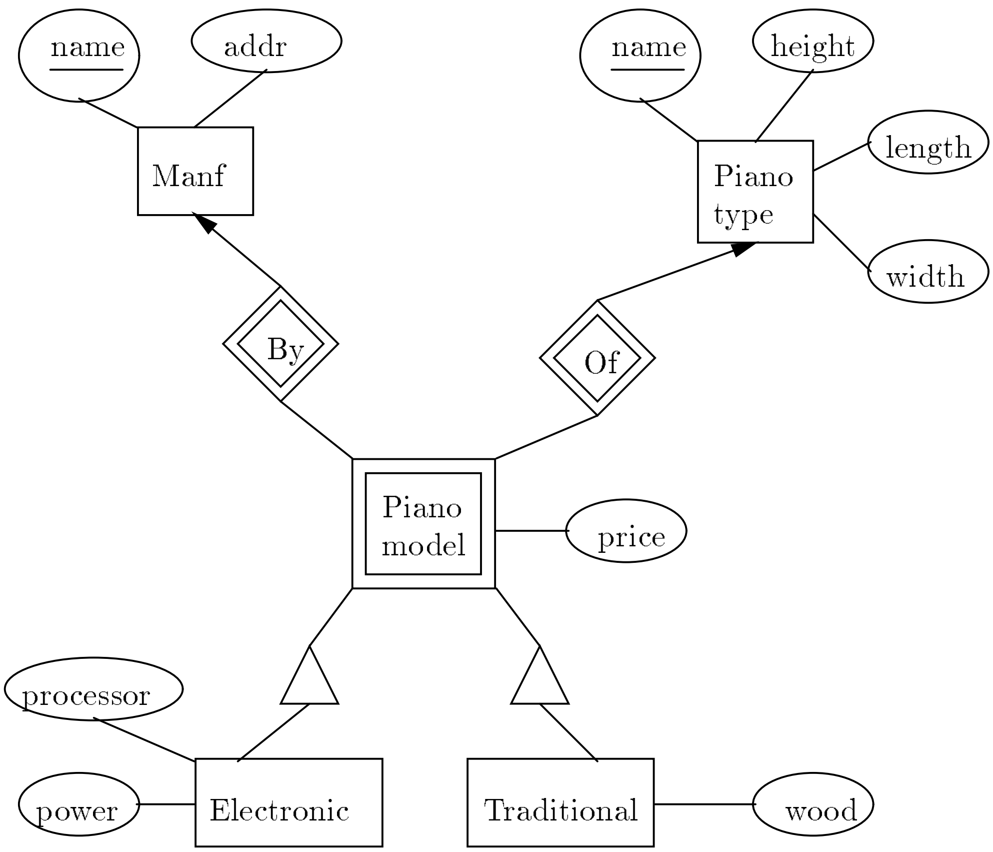
I acknowledge and accept the honor code.

____

Print your name here: ____

1	2	3	

Problem 1: (20 points) The figure below is an E/R diagram about pianos, their manufacturers and types. A typical manufacturer could be Steinway or Casio. Typical names of types are Grand, Baby-Grand, Spinet, and (electronic) Keyboard. Pianos are either electronic or “traditional”; some could be both, and some could be neither. For traditional pianos we record the wood from which it is made, and for electronic pianos, we record the processor used and the power.



a) In this diagram, is it possible to represent the fact that Steinway makes two different models of baby-grand piano? If so, explain how. If not, explain how you would modify the diagram to make it possible.

b) Convert this E/R diagram to a relational database schema, using the "E/R" approach to handling the isa-hierarchy.

c) Convert the same E/R diagram to a database schema using the "object-oriented" approach to handling the isa-hierarchy.

Problem 2: (20 points) Consider the relation $R(A, B, C, D, E)$ with functional dependencies $A \to B$, $B \to C$, $C \to A$, $D \to E$, and $E \to D$.

a) What are all the keys for R?

b) How many superkeys are there? _______ Justify your answer for partial credit (optional).

c) One possible decomposition of R is into $R_1(A, B)$ and $R_2(A, C, D, E)$. Give a basis for the FD's that hold in each of these relations, i.e., for each relation a set of dependencies with singleton right sides from which all true FD's can be deduced, using only the attributes of that relation.

For $R_1(A, B)$: ____________________________________

For $R_2(A, C, D, E)$: ____________________________________

d) Give a counterexample (instance of R) that shows that the multivalued dependency $AB \twoheadrightarrow CD$ need not hold for R with the given FD's.

A	B	C	D	E

Problem 3: (35 points) *Warning*: Be sure you write in SQL (the SQL2 standard, not Oracle SQL) when asked to and in relational algebra when asked to. You will not receive credit for using the wrong language. When writing in relational algebra, you may use the "named intermediates" style of breaking the expression into several small steps.

This question revolves around the following two relations:

```
Articles(ID, dateline, headline, author, text)
Keywords(ID, keyword)
```

The intent of the first is that each tuple represents a news article: a unique ID for that article, the day written, the headline of the article, the author, and the text of the article. The second relation gives zero or more keywords for each article.

a) If we declare a schema for `Articles`, we might want to enforce the constraint that `dateline author` $\rightarrow$ `ID`; that is, no one can write more than one article in one day. Show how to write this constraint *in SQL* as a tuple-based check.

b) Find the headlines of articles for which "Arafat" is a keyword. Write *in SQL*.

c) Find the ID's of articles that have at least three keywords. Write *in relational algebra*.

d) Find the ID's of articles in which "Pol Pot" appears somewhere in the text field. Write *in SQL*.

e) Print a table giving, for each author and for each keyword of three or more of that author's articles, the earliest dateline among this set of articles (i.e., the set of articles by this author and with this keyword). Write *in SQL*.

f) Find all the authors who never wrote an article with keyword "Milosevic." Write *in relational algebra.*

g) Modify the `Articles` relation so that any article with a `NULL` text is given the headline of that article as its text. Write *in SQL*.