CS 2322 Laboratory Problem Set 7

Extra Credit Problem Set

These problems are to be done on an individual basis following the Trinity University Academic Integrity Policy or Trinity University Honor Code.

Academic Integrity and Honor Code

All students are covered by a policy that prohibits dishonesty in academic work. The Academic Integrity Policy (AIP) covers all students who entered Trinity before the Fall of 2004. The Academic Honor Code covers all those who entered the Fall of 2004 or later. The Integrity Policy and the Code share many features: each asserts that the academic community is based on honesty and trust; each contains the same violations; each provides for a procedure to determine if a violation has occurred and what the punishment will be; each provides for an appeal process. The main difference is that the faculty implements the AIP while the Honor Code is implemented by the Academic Honor Council. Under the Academic Integrity Policy, the faculty member determines whether a violation has occurred as well as the punishment for the violation (if any) within certain guidelines. Under the Honor Code, a faculty member will (or a student may) report an alleged violation to the Academic Honor Council. It is the task of the Council to investigate, adjudicate, and assign a punishment within certain guidelines if a violation has been verified. Students who are under the Honor Code are required to pledge all written work that is submitted for a grade: On my honor, I have neither given nor received any unauthorized assistance on this work and heir signature. The pledge may be abbreviated pledged with a signature.

Laboratory problems should be submitted electronically (e-mail to cs2322@cs.trinity.edu) on or before the due date and should contain a problem write-up, source code to any programs and data sets used in solving the problem. The submitted files should be ASCII text files having Unix end-of-line characters (please convert all Windows and Mac text files to Unix format–I have found that Emacs seems to do a reasonable job of such conversions). If several files need to be submitted, put them in a directory having name *your-last-name-problem-set-number* and create a tar archive of this file system and attach it to your e-mail problem submission.

Modeling Relational Databases

The following definitions provide a simple relational database capability. Try to develop solutions to one or two of the problems posed at the end of the following set of J definitions.

```
NB. Relational Database Definitions

first =: {.
rest =: }.
open =: >
box =: <
head =: open@first
rank =: "
bond =: &
link =: ;

NB. A relation is a collection of tuples which we represent as a table.

csg =: ('course';'studentid';'grade');('cs101';12345;'A');('cs101';67890;'B');('ee200';12345;'C');('ee200';22222;'B+');('cs101';333333;'A-');box('
```

```
NB. In the definition of the relation csg we have
NB. incorporated additional information attributes (a set of column names called
NB. scheme) at the beginning of our representation for a relation.

head csg NB. course_studentid_grade scheme
rest csg NB. course_studentid_grade relation

NB. The order of tuples in a relation is not significant.
NB. The order of columns in a relation is not significant.
NB. We do assume that column names are distinct.

NB. access column of a relation by name
NB. relation col 'column_name'

col =: dyad def '((head x.) i. box y.) bond from rank 1 open rest x.'

NB. access row of a relation by row_index
NB. relation row row_index

row =: dyad def 'open y. from rest x.'

NB. Some other sample relations.

snap =: ('studentid';'name';'address';'phone');(12345;'C. Brown';'12 Apple St.';'555-1234');(67890;'L. Van Pelt';'34 Pear Ave.';'555-5678');box(22

cp =: ('course';'prerequisite');('cs101';'cs100');('ee200';'ee005');('ee200';'cs100');('cs120';'cs101');('cs121';'cs120');('cs205';'cs101');('cs20

cdh =: ('course';'day';'hour');('cs101';'M';'9AM');('cs101';'W';'9AM');('cs101';'F';'9AM');('ee200';'Tu';'10AM');('ee200';'W';'1PM');box('ee200';'

cr =: ('course';'room');('cs101';'Turing Aud.');('ee200';'25 Ohm Hall');box('ph100';'Newton Lab.')

NB. A database is a collection of relations

NB. Problem 1.  Implement relation insert tuple
NB. Problem 2.  Implement relation delete tuple_spec
NB.             where '*' in a tuple_spec means match anything.
NB. Problem 3.  Implement relation lookup tuple_spec
NB.             where '*' in a tuple_spec means match anything.

NB. A key for a relation is a set of one or more attributes such that
NB. under no circumstances will the relation have two tuples whose
NB. values agree in each column headed by a key attribute.
```

This set of definitions is available in the class files (`/users/jhowland/cs2322j/relational.ijs` directory for this course.

Problem Set 7 Solution [HTML] [PS] [PDF]