Harvey Mudd College
Computer Science 133
Database and Knowledgebase Systems
Fall Semester 2001

Assignment #2 – Relational Algebra and RDB

Due 5:00pm, Friday September 28, 2001

This homework involves using an implementation of the relational algebra called `RDB` on `turing`. The executable of `RDB` is located in the directory `/cs/cs131/rdb/`. The databases you need to manipulate for this assignment are in the `samples` subdirectory of that directory. Documentation on `RDB` is in the file `rdbman.txt` in the `rdb` directory and also available on the course home page.

In submitting answers, provide a copy of interactive sessions in which the various values and functions are defined, and in which results are displayed. To do this, you should either run your session inside of an emacs shell buffer or use the unix command `script`, which saves an interactive session in a file that you specify. Please submit readable output; I do not want to see your mistakes, only final answers. It will help to do a trial run, or to edit the script file once you are done. You may also use a colored pen (or alternate fonts) to beautify your output and to add a few appropriate comments – in particular, to indicate in a conspicuous way where each question or part question begins and to explain the use and meaning of identifiers where appropriate. If you choose to add comments into the text file without using a distinct font, they should be marked with a percent sign (`%`) at the beginning of the line.

Note: `RDB` allows you to construct relations with more than one attribute with the same name, and to project over one of those attributes using a numbering convention, e.g.

```
rdb> let temp= join owners with pets;
relation temp bound
rdb> project temp over pname:1;
```

However, it doesn't allow you to use numbered attributes in comparison expressions,

```
rdb> restrict temp with pname:1='Fido';
?error in parsing ".....strict temp with pname : "
expecting ";"
```

You should therefore use the `rename` command to avoid this sort of problem.

1. The first part of the homework uses `RDB` with a simple animal shelter database. It is stored in the file `/cs/cs133/rdb/samples/as.rdb` and contains two relations: **pets** and **owners**.

   The **pets** relation has six attributes: pname (pet name), ptype (i.e. `cat` or `dog`), breed, reason (why this pet is in shelter, such as `stray`, owner `moved`, `abandoned`, etc.), fixed (i.e. spayed or neutered), and adopted. The last two fields are either `yes` or `no`. You may assume that {pname, ptype} is a key for **pets**.

   The **owners** relation has five attributes: oname, phone, pname (name of adopted pet), ptype and fixed (the pet, not the owner!). The key for **owners** is {oname, pname, ptype}.

   For each question, you are to give the Relational Algebra answer as well as the `RDB` answer. The RA answer may be written in pen, or using a suitable math font. Please group the two answers to each question together, with the RA coming first. Do not give all the RA answers on a separate sheet.

   The first four queries are simple, and require access to a single relation. The rest require access to multiple relations (remember your binary operators). You may break an answer up into a series of queries, using `it` to refer to the previous result. You may also store an intermediate answer in a new named relation (e.g. `let temp = ...`).

   Use the animal shelter database to construct the following relations:

   (a) All pets adopted by John and their types.

   (b) Names of all cats in the database.

   (c) Names of all adopted cats in the database.

   (d) Telephone numbers of owners who adopted cats.

   (e) Names of owners who adopted short-hair stray cats.

   (f) Names and telephone numbers of owners whose pets are not fixed yet.

   (g) Names of owners who adopted two or more pets of the same type.

   (h) Print the number of cats that John has adopted. (Hint: use the `reduce` operator described in the `RDB` manual. Also, this does not have a complete answer in relational algebra. Give as much as you can for the RA answer.)

   (i) Names of owners who adopted both cats and dogs.

   (j) Names of owners who adopted at least one of each type of animal in the shelter. Make sure this will work even if we add new pet types to the database.

   (k) Observe that the **pets** and **owners** relations may not agree on the fixed attribute if only one relation is updated when owners inform the shelter that adopted pets were spayed/neutered. Assuming that the **owners** table has the correct information, update the **pets** relation to make it consistent with the **owners** relation using the set operations difference and union.

2. At the RDB prompt, create a relation **parent**(pname,cname), indicating that pname is the parent of cname in some family. The relation should contain the following tuples:

   (Anne, Boris), (Boris, Charles), (Boris, Emily), (Charles, Darlene), (Darlene, Fred).

   (a) Give Relational Algebra and RDB constructions for the following relations:
      i. **grandparent**(gpname, gcname)
      ii. **greatgrand**(ggpname, ggcname)

   (b) Generalize this to define the function **anscestor**(**parent_relation**), which, when called with a table with columns pname and cname, will return a relation instance of the schema {anc, dec}, where anc is an ancestor of dec (who is in turn a decendent of anc), built from that relation.