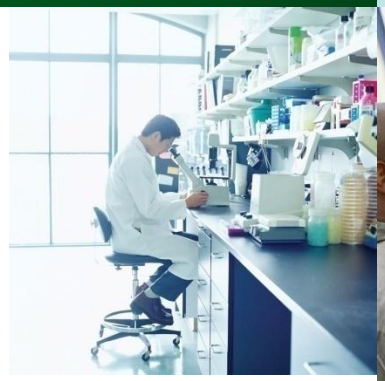
Hash Tables



Dieter Ayers

Surveillance Biostatistician

Population & Public Health

Provincial Health Services Authority

Provincial Health Services Authority
Province-wide solutions. Better health.

The motivation

The CCHS is a cross-sectional survey that collects information related to health status, health care utilization and health determinants for the Canadian population. It relies upon a large sample of respondents and is designed to provide reliable estimates at the health region level. The CCHS samples roughly 65,000 respondents annually. In British Columbia, we have roughly 7,500 respondents annually.

In 2008/2009 we commissioned an oversample, to allow for estimation at a smaller geographical scale. This sample had 22,500 respondents, and 506 variables.

What can hash tables do?

They can merge datasets on a common key (without sorting)!

Join three datasets in a single step (still without sorting)!

Conditional joins on multiple tables!

Recursive Lookups!

Outputting N datasets at runtime, when N is unknown!

And they do it fast!

What is a hash table?

Hash Tables (from wikipedia):

In computer science, a **hash table** or **hash map** is a data structure that uses a hash function to map identifying values, known as keys (e.g., a person's name), to their associated values (e.g., their telephone number). Thus, a hash table implements an associative array.

In SAS:

A hash object is an in-memory lookup table accessible from the data step. It consists of:

1) keys,
2) data.

from the log

```
3496  data tmf;
3497      set sasdat.tmf;
3498  run;

NOTE: There were 126345 observations read from the data set SASDAT.TMF.
NOTE: The data set WORK.TMF has 126345 observations and 29 variables.
NOTE: DATA statement used (Total process time):
      real time          7.73 seconds
      cpu time           0.35 seconds

3500  data cchs;
3501  set sasdat.bcbi;
3502  run;

NOTE: There were 22527 observations read from the data set SASDAT.BCBI.
NOTE: The data set WORK.CCHS has 22527 observations and 507 variables.
NOTE: DATA statement used (Total process time):
      real time          24.82 seconds
      cpu time           0.70 seconds
```

data step processing

```
data datastepmerged;
    merge cchs (in=survey) tmf(keep=pstl latitude longitude);
    by pstl;
    if survey;
run;
```

from the log

ERROR: BY variables are not properly sorted on data set WORK.CCHS.

NOTE: The SAS System stopped processing this step because of errors.

NOTE: There were 7 observations read from the data set WORK.CCHS.

NOTE: There were 16856 observations read from the data set WORK.TMF.

WARNING: The data set WORK.DATASTEPMERGED may be incomplete.
When this step was stopped there were 5 observations and 509 variables.

NOTE: DATA statement used (Total process time):

real time 0.14 seconds

cpu time 0.14 seconds

data step processing

```
proc sort data=tmf;
    by pstl;
proc sort data=cchs;
    by pstl;
run;

data datastepmerged;
    merge cchs (in=survey) tmf(keep=pstl latitude longitude);
    by pstl;
    if survey;
run;
```

from the edited log

```
3532  proc sort data=tmf;
NOTE: PROCEDURE SORT used (Total process time):
      real time           3.54 seconds
      cpu time            0.24 seconds

3534  proc sort data=cchs;
NOTE: PROCEDURE SORT used (Total process time):
      real time           7.82 seconds
      cpu time            0.89 seconds

3540  data datastepmerged;
3541      merge cchs (in=survey) tmf(keep=pstl latitude longitude);
3542      by pstl;
3543      if survey;
3544  run;

NOTE: DATA statement used (Total process time):
      real time           5.18 seconds
      cpu time            0.64 seconds
```

hashing

```
data hashmerged;
    set cchs;
    if _N_ = 1 then do;
    length latitude longitude $8.;
    declare hash latlong (dataset:'tmf', hashexp:16);
            latlong.definekey('pstl');
            latlong.definedata('pstl', 'latitude','longitude');
            latlong.definedone();
    end;
    if latlong.find() then;
    *if  latlong.find() = 0 then output;
run;
```

from the log

```
3604  data hashmerged;
3605      set cchs;
3606      if _N_ = 1 then do;            *don't initialize more than 1 hash;
3607      length latitude longitude $8.;  *must declare types of new variables;
3608      declare hash latlong (dataset:'tmf'); *use this data;
3609          latlong.definekey('pstl');
3610          latlong.definedata('pstl', 'latitude','longitude');
3611          latlong.definedone();
3612      end;
3613      if latlong.find() then;
3614      *if ~latlong.find() then output;
3615  run;
```

```
NOTE: There were 126345 observations read from the data set WORK.TMF.
NOTE: There were 22527 observations read from the data set WORK.CCHS.
NOTE: The data set WORK.HASHMERGED has 22527 observations and 509 variables.
NOTE: DATA statement used (Total process time):
      real time          3.35 seconds
      cpu time           0.61 seconds
```

```
data hashmerged;
    set cchs;
    if _N_ = 1 then do;
    length latitude longitude $8.;
    declare hash latlong (dataset:'tmf', hashexp:16);
            latlong.definekey('pstl');
            latlong.definedata('pstl', 'latitude','longitude');
            latlong.definedone();
    end;
    if latlong.find() then;
    *if  latlong.find() = 0 then output;
run;
```

Some other methods

Add

Check

Clear

Delete

Equals

Find_next

Find_prev

First

Has_prev

Item_size

Last

Has_next

Next

Num_items

Output

Prev

Ref

Remove

Removedup

Setcur

Sum

Sumdup

N individual data sets

```
data lha1;
    set cchs;
    if geodlha eq 1;
data lha2;
    set cchs;
    if geodlha eq 2;
data lha3;
    set cchs;
    if geodlha eq 3;
data lha4;
    set cchs;
    if geodlha eq 4;
data lha5;
    set cchs;
    if geodlha eq 5;
```

N individual data sets

```
data _null_;
    if 0 then set cchs;
    declare hash groups (ordered: 'a');
    groups.definekey('_n_');
    groups.definedata('GEODPC', 'HWTDBMI', 'INCDPER', 'PACDPAI');
    groups.definedone();

    do _n_ = 1 by 1 until (last.geodlha);
        set cchs;
        by geodlha;
        groups.add();
        end;
    groups.output (dataset: compress('LHA'||geodlha));
run;
```

References

I cut my processing time by 90% using hash tables - You can do it too!
Jennifer K. Warner-Freeman
www.nesug.org/proceedings/nesug07/bb/bb16.pdf

Data Step Hash Objects as Programming Tools. Paul M. Dorfman
http://www2.sas.com/proceedings/sugi30/236-30.pdf

How Do I Love Hash Tables? Let Me Count The Ways! Judy Loren
http://www2.sas.com/proceedings/forum2008/029-2008.pdf

Getting Started with the DATA Step Hash Object. Jason Secosky, Janice Bloom
http://support.sas.com/rnd/base/datastep/dot/hash-getting-started.pdf