examples/copy_record.pql by Pequel

sample@youraddress.com

Copy Record Example Script

Table of Contents Copy Record Example Script

SCRIPT NAME	1
DESCRIPTION	1
1. PROCESS DETAILS	1
1.1 LOCATION	1
Description	1
Derived Input Field Evaluation	1
1.2 DESCRIPTION	1
Description	1
Derived Input Field Evaluation	1
1.3 SALES_TOTAL	1
Description	1
2. CONFIGURATION SETTINGS	2
2.1 prefix	2
2.2 pequeldoc	2
2.3 detail	2
2.4 script_name	2
2.5 input_file	2
2.6 optimize	2
2.7 doc_title	2
2.8 doc_email	2
2.9 doc_version	2
3. TABLES	3
3.1 LOC_DESCRIPT	3
Data	3
4. TABLE INFORMATION SUMMARY	4
4.1 Table List Sorted By Table Name	4
5. EXAMPLES/COPY_RECORD.PQL	5
options	5
init table	5
input section	5
copy record(pequel:copy_record_WA.pql)	5
copy record(pequel:copy_record_SA.pql)	5
copy record(pequel:copy_record_NSW.pql)	5
copy record(pequel:copy_record_VIC.pql)	5
copy record(pequel:copy_record_NT.pql)	5
filter	5
group by	5
output section	5
sort output	5
6. PEQUEL GENERATED PROGRAM	6
7. ABOUT PEQUEL	15
COPYRIGHT	15

SCRIPT NAME

examples/copy_record.pql

DESCRIPTION

1. PROCESS DETAILS

Input records are read from chain_pequel_pt1.pql. The input record contains **3** fields. Fields are delimited by the '|' character.

Output records are written to standard output. The output record contains **3** fields. Fields are delimited by the '|' character.

Input records are eliminated (filtered) unless LOCATION eq 'WA' || LOCATION eq 'SA' || LOCATION eq 'NSW' || LOCATION eq 'VIC' || LOCATION eq 'NT'.

Input records are **grouped** by the input field **LOCATION** (string).

1.1 LOCATION

Output Field

Description

Set to input field LOCATION_DESC

Derived Input Field Evaluation

=> %LOC_DESCRIPT(LOCATION)

1.2 DESCRIPTION

Output Field

Description

Set to input field **DESCRIPTION**

Derived Input Field Evaluation

=> 'State Total'

1.3 SALES_TOTAL

Output Field

Description

Sum aggregation on input field **SALES_TOTAL**.

2. CONFIGURATION SETTINGS

2.1 prefix

directory pathname prefix.: examples

2.2 pequeldoc

generate pod / pdf pequel script Reference Guide.: pdf

2.3 detail

Include Pequel Generated Program chapter in Pequeldoc: 1

2.4 script_name

script filename: examples/copy_record.pql

2.5 input file

input data filename: chain_pequel_pt1.pql

2.6 optimize

optimize generated code.: 1

2.7 doc_title

document title.: Copy Record Example Script

2.8 doc_email

document email entry.: sample@youraddress.com

2.9 doc_version

document version for pequel script.: 2.3

3. TABLES

3.1 LOC_DESCRIPT

Table Type: Iocal

Data

NSW — New South Wales

WA — Western Australia

SA — South Australia

NT — Northern Territory

 $\mathsf{QLD}-\mathsf{Queensland}$

VIC — Victoria

4. TABLE INFORMATION SUMMARY

4.1 Table List Sorted By Table Name LOC_DESCRIPT — *1 (local)*

5. EXAMPLES/COPY_RECORD.PQL

options

```
prefix(examples)
pequeldoc(pdf)
detail(1)
script_name(examples/copy_record.pql)
input_file(chain_pequel_pt1.pql)
optimize(1)
doc_title(Copy Record Example Script)
doc_email(sample@youraddress.com)
doc_version(2.3)
```

init table

```
LOC_DESCRIPT NSW New South Wales
LOC_DESCRIPT WA Western Australia
LOC_DESCRIPT SA South Australia
LOC_DESCRIPT NT Northern Territory
LOC_DESCRIPT QLD Queensland
LOC_DESCRIPT VIC Victoria
```

input section

```
LOCATION
PRODUCT_CODE
SALES_TOTAL
LOCATION_DESC => %LOC_DESCRIPT(LOCATION)
DESCRIPTION => 'State Total'
```

copy record(pequel:copy_record_WA.pql)

LOCATION eq 'WA'

copy record(pequel:copy_record_SA.pql)

LOCATION eq 'SA'

copy record(pequel:copy_record_NSW.pql)

LOCATION eq 'NSW'

copy record(pequel:copy_record_VIC.pql)

LOCATION eq 'VIC'

copy record(pequel:copy_record_NT.pql)

LOCATION eq 'NT'

filter

```
 \texttt{LOCATION eq 'WA' || LOCATION eq 'SA' || LOCATION eq 'NSW' || LOCATION eq 'VIC' || LOCATION eq 'NT' ||
```

group by

LOCATION string

output section

```
string LOCATION LOCATION_DESC string DESCRIPTION DESCRIPTION decimal SALES_TOTAL sum SALES_TOTAL
```

sort output

```
LOCATION string SALES_TOTAL numeric des
```

6. PEQUEL GENERATED PROGRAM

```
#!/usr/bin/perl
\# vim: syntax=perl ts=4 sw=4
#Generated By: pequel Version 2.4-5, Build: Wednesday November 16 21:56:42 GMT 2005
                     : http://sourceforge.net/projects/pequel/
#Script Name : copy_record.pql
#Created On : Wed Nov 16 14:00:48 2005
#Perl Version: /usr/bin/perl 5.6.1 on solaris
#For
#Options:
#prefix(examples) directory pathname prefix.
#pequeldoc(pdf) generate pod / pdf pequel script Reference Guide.
#detail(1) Include Pequel Generated Program chapter in Pequeldoc
#script_name(examples/copy_record.pql) script filename
#input_file(chain_pequel_pt1.pql) input data filename
#optimize(1) optimize generated code.
#doc_title(Copy Record Example Script) document title.
\verb|#doc_email(sample@youraddress.com)| document email entry.
\#doc\_version(2.3) document version for pequel script.
#-----
                                                                                         use strict;
use Fcntl ':flock';
use constant \_I\_LOCATION
                                                    => int
                                                                    0;
use constant _I_PRODUCT_CODE
                                                    => int.
                                                                     1;
use constant _I_SALES_TOTAL
                                                     => int
                                                                      2:
use constant _I_LOCATION_DESC => int
                                                                      3;
use constant _I_DESCRIPTION
                                                    => int
                                                                      4;
use constant _O_LOCATION
                                                    => int.
                                                                      1;
use constant _O_DESCRIPTION
                                                  => int
                                                                      2;
use constant _O_SALES_TOTAL
                                                    => int
                                                                      3;
use constant _T_LOC_DESCRIPT_FLD_1
                                                           => int
                                                                            0;
use constant _I_LOC_DESCRIPT_LOCATION_FLD_KEY => int
                                                                                           5;
use constant _I_LOC_DESCRIPT_LOCATION_FLD_1 => int
local $\="\n";
local $,="|";
print STDERR '[examples/copy_record.pql ' . localtime() . "] Init";
use constant VERBOSE => int 10000;
use constant LAST ICELL => int 4;
my @I VAL;
my @O_VAL;
my $_inprecs=0;
my $key__I_LOCATION;
my $previous_key__I_LOCATION = undef;
foreach my $f (1..3) { $O_VAL[$f] = undef; }
\verb|my $$_{TABLE\_LOC\_DESCRIPT = \&InitLookupLOC\_DESCRIPT; \# ref to $$LOC\_DESCRIPT hash $$ and $$_{TABLE\_LOC\_DESCRIPT = \&InitLookupLOC\_DESCRIPT; \# ref to $$$LOC\_DESCRIPT hash $$ and $$_{TABLE\_LOC\_DESCRIPT = \&InitLookupLOC\_DESCRIPT; \# ref to $$$$LOC\_DESCRIPT hash $$$ and $$_{TABLE\_LOC\_DESCRIPT = \&InitLookupLOC\_DESCRIPT; \# ref to $$$$$ and $$_{TABLE\_LOC\_DESCRIPT = \&InitLookupLOC\_DESCRIPT = \&In
if (open(READ_CHAIN_PEQUEL_PT1, '-|') == 0) # Fork -- read from child
{
      &p_read_chain_pequel_pt1::read_chain_pequel_pt1;
      exit(0);
}
open(STDOUT, '|-', q{sort -t'|' -y -k 1,1 -k 3nr,3nr 2>/dev/null});
if (open(COPY_INPUT_COPY_RECORD_WA, '|-') == 0) # Fork -- write to child
      &p_copy_input_copy_record_wa::copy_input_copy_record_wa;
      exit(0);
}
if (open(COPY_INPUT_COPY_RECORD_SA, '|-') == 0) # Fork -- write to child
{
      &p_copy_input_copy_record_sa::copy_input_copy_record_sa;
      exit(0);
if (open(COPY_INPUT_COPY_RECORD_NSW, '|-') == 0) # Fork -- write to child
{
      &p copy input copy record nsw::copy input copy record nsw;
      exit(0);
}
if (open(COPY_INPUT_COPY_RECORD_VIC, '|-') == 0) # Fork -- write to child
      &p_copy_input_copy_record_vic::copy_input_copy_record_vic;
      exit(0);
if (open(COPY_INPUT_COPY_RECORD_NT, '|-') == 0) # Fork -- write to child
```

```
&p_copy_input_copy_record_nt::copy_input_copy_record_nt;
   exit(0);
print STDERR '[examples/copy_record.pql ' . localtime() . "] Start";
use Benchmark;
my Sbenchmark start = new Benchmark;
while (<READ_CHAIN_PEQUEL_PT1>)
    ++$ inprecs;
   print STDERR '[examples/copy_record.pql ' . localtime() . "] $_inprecs records." if ($_inprecs % VERBOSE =
   chomp;
   @I_VAL = split("[|]", $_);
next unless ($I_VAL[_I_LOCATION] eq 'WA' || $I_VAL[_I_LOCATION] eq 'SA' || $I_VAL[_I_LOCATION] eq 'NSW' || $I_VAL[_I_LOCATION] eq 'VIC' || $I_VAL[_I_LOCATION] eq 'NT');
   if (($I_VAL[_I_LOCATION] eq 'WA'))
       print COPY INPUT COPY RECORD WA $ ;
   if (($I_VAL[_I_LOCATION] eq 'SA'))
    {
       print COPY_INPUT_COPY_RECORD_SA $_;
    }
    if (($I_VAL[_I_LOCATION] eq 'NSW'))
       print COPY_INPUT_COPY_RECORD_NSW $_;
    }
    if (($I_VAL[_I_LOCATION] eq 'VIC'))
       print COPY_INPUT_COPY_RECORD_VIC $_;
    if (($I_VAL[_I_LOCATION] eq 'NT'))
       print COPY_INPUT_COPY_RECORD_NT $_;
    $key__I_LOCATION = $I_VAL[_I_LOCATION];
    if (!defined($previous_key__I_LOCATION))
        $previous_key__I_LOCATION = $key__I_LOCATION;
    elsif ($previous_key__I_LOCATION ne $key__I_LOCATION)
        flock(STDOUT, LOCK_EX);
       print STDOUT
           $0_VAL[_O_LOCATION],
            $0_VAL[_O_DESCRIPTION],
            $0_VAL[_O_SALES_TOTAL]
        flock(STDOUT, LOCK_UN);
        $previous_key__I_LOCATION = $key__I_LOCATION;
        @O_VAL = undef;
    $O_VAL[_O_LOCATION] = $I_VAL[_I_LOCATION_DESC];
    $I_VAL[_I_DESCRIPTION] = 'State Total'
    $0_VAL[_0_DESCRIPTION] = $I_VAL[_I_DESCRIPTION];
    $0_VAL[_O_SALES_TOTAL] += $1_VAL[_I_SALES_TOTAL] unless ($1_VAL[_I_SALES_TOTAL] eq '');
flock(STDOUT, LOCK_EX);
print STDOUT
   $0_VAL[_O_LOCATION]
    $0_VAL[_O_DESCRIPTION];
   $0_VAL[_O_SALES_TOTAL]
flock(STDOUT, LOCK_UN);
close(COPY_INPUT_COPY_RECORD_NT);
close(COPY_INPUT_COPY_RECORD_VIC);
close(COPY_INPUT_COPY_RECORD_NSW);
close(COPY_INPUT_COPY_RECORD_SA);
close(COPY_INPUT_COPY_RECORD_WA);
close(STDOUT);
close(READ_CHAIN_PEQUEL_PT1);
print STDERR '[examples/copy_record.pql ' . localtime() . "] $_inprecs records.";
my $benchmark_end = new Benchmark;
my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
print STDERR '[examples/copy_record.pql ' . localtime() . "] Code statistics: @{[timestr($benchmark_timediff)]
```

```
#+++++ Table LOC_DESCRIPT --> Type :ETL::Pequel::Type::Table::Local +++++
sub InitLookupLOC DESCRIPT
   my %_TABLE_LOC_DESCRIPT;
   %_TABLE_LOC_DESCRIPT =
        'NSW' => 'New South Wales',
       'NT' => 'Northern Territory',
       'QLD' => 'Queensland'
       'SA' => 'South Australia',
       'VIC' => 'Victoria',
       'WA' => 'Western Australia'
   return \% TABLE LOC DESCRIPT;
}
{
   package p_copy_input_copy_record_sa;
   sub copy_input_copy_record_sa
#
    !/usr/bin/perl
   vim: syntax=perl ts=4 sw=4
    Generated By: pequel Version 2.4-5, Build: Wednesday November 16 21:56:42 GMT 2005
               : http://sourceforge.net/projects/pequel/
    Script Name : copy_record_SA.pql
    Created On : Wed Nov 16 14:00:43 2005
    Perl Version: /usr/bin/perl 5.6.1 on solaris
   Options:
        optimize(1) optimize generated code.
        hash(1) Generate in memory. Input data can be unsorted.
        doc_title(Copy Record Example Script) document title.
        doc_email(sample@youraddress.com) document email entry.
       doc_version(2.3) document version for pequel script.
       use strict;
       use Fcntl ':flock';
       use constant _I_PRODUCT_CODE => int use constant _I_SALES TOTAY
                                                0;
                                               1;
       use constant _I_SALES_TOTAL => int
use constant _I_LOCATION_NAME => int
       use constant _O_LOCATION_NAME => int
use constant _O_PRODUCT_CODE => int
       use constant _O_SALES_TOTAL
                                      => int
       local \= \n'';
       local $,="|";
       print STDERR '[examples/copy_record_SA.pql ' . localtime() . "] Init";
       use constant VERBOSE => int 10000;
       use constant LAST_ICELL => int 3;
       my @I_VAL;
       my %O_VAL;
       my $key;
       my $_inprecs=0;
       print STDERR '[examples/copy_record_SA.pql ' . localtime() . "] Start";
       use Benchmark;
       my $benchmark_start = new Benchmark;
       while (<STDIN>)
           ++$_inprecs;
           print STDERR '[examples/copy_record_SA.pql ' . localtime() . "] $_inprecs records." if ($_inprecs
% VERBOSE == 0);
           chomp;
           @I_VAL = split("[|]", $_);
           $key = ( $I_VAL[_I_PRODUCT_CODE] );
           $I_VAL[_I_LOCATION_NAME] = 'South Australia';
           $O_VAL{$key}{_O_LOCATION_NAME} = $I_VAL[_I_LOCATION_NAME];
           $0_VAL{$key}{_O_PRODUCT_CODE} = $I_VAL[_I_PRODUCT_CODE];
           $O_VAL{$key}{_O_SALES_TOTAL} += $I_VAL[_I_SALES_TOTAL] unless ($I_VAL[_I_SALES_TOTAL] eq '');
       foreach $key (sort keys %O_VAL)
           flock(STDOUT, LOCK_EX);
           print STDOUT
               $O_VAL{$key}{_O_LOCATION_NAME},
               $0_VAL{$key}{_O_PRODUCT_CODE},
               O_VAL{skey}{O_SALES\_TOTAL}
           flock(STDOUT, LOCK_UN);
       }
```

```
close(STDIN);
            print STDERR '[examples/copy_record_SA.pql ' . localtime() . "] $_inprecs records.";
            my $benchmark end = new Benchmark;
            my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
             \texttt{print STDERR '[examples/copy\_record\_SA.pql ' . local time() . "] Code statistics: @\{[timestr(\$benchmark of the statistics] and the statistics] are statistically considered as a statistic of the statistic o
 _timediff)]}";
                                 }
}
{
      package p_copy_input_copy_record_wa;
      sub copy_input_copy_record_wa
       !/usr/bin/perl
#-+
       vim: syntax=perl ts=4 sw=4
Generated By: pequel Version 2.4-5, Build: Wednesday November 16 21:56:42 GMT 2005
                          : http://sourceforge.net/projects/pequel/
       Script Name : copy_record_WA.pql
       Created On : Wed Nov 16 14:00:41 2005
       Perl Version: /usr/bin/perl 5.6.1 on solaris
       For
                         Options:
             optimize(1) optimize generated code.
             hash(1) Generate in memory. Input data can be unsorted.
             doc_title(Copy Record Example Script) document title.
             doc_email(sample@youraddress.com) document email entry.
            doc_version(2.3) document version for pequel script.
use strict;
            use Fcntl ':flock';
            use constant \_I\_LOCATION
                                                             => int
            use constant _I_PRODUCT_CODE => int
            use constant _I_SALES_TOTAL
                                                              => int
           use constant _I_LOCATION_NAME => int
use constant _O_LOCATION_NAME => int
                                                              => int
            1;
            local \= \n'';
            local $,="|";
            print STDERR '[examples/copy_record_WA.pql ' . localtime() . "] Init";
            use constant VERBOSE => int 10000;
            use constant LAST_ICELL => int 3;
            my @I_VAL;
            my %O_VAL;
            my $key;
            my $_inprecs=0;
            print STDERR '[examples/copy_record_WA.pql ' . localtime() . "] Start";
            use Benchmark;
            my $benchmark_start = new Benchmark;
            while (<STDIN>)
                   ++$ inprecs;
                  print STDERR '[examples/copy_record_WA.pql ' . localtime() . "] $_inprecs records." if ($_inprecs
% VERBOSE == 0);
                  chomp
                  @I_VAL = split("[|]", $_);
                  $key = ( $I_VAL[_I_PRODUCT_CODE] );
                  $I_VAL[_I_LOCATION_NAME] = 'Western Australia';
                  $0_VAL{$key}{_O_LOCATION_NAME} = $I_VAL[_I_LOCATION_NAME];
                   $O_VAL{$key}{_O_PRODUCT_CODE} = $I_VAL[_I_PRODUCT_CODE];
                  $O_VAL{$key}{_O_SALES_TOTAL} += $I_VAL[_I_SALES_TOTAL] unless ($I_VAL[_I_SALES_TOTAL] eq '');
            foreach $key (sort keys %O_VAL)
                  flock(STDOUT, LOCK_EX);
                  print STDOUT
                        O_VAL{skey}_{O_LOCATION_NAME},
                         O_VAL{skey}{O_PRODUCT_CODE},
                         $0_VAL{$key}{_O_SALES_TOTAL}
                  flock(STDOUT, LOCK_UN);
            }
            close(STDIN);
            print STDERR '[examples/copy_record_WA.pql ' . localtime() . "] $_inprecs records.";
            my $benchmark_end = new Benchmark;
            my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
            print STDERR '[examples/copy_record_WA.pql ' . localtime() . "] Code statistics: @{[timestr($benchmark
_timediff)]}";
```

```
}
{
   package p read chain pequel pt1;
   sub read_chain_pequel_pt1
    !/usr/bin/perl
vim: syntax=perl ts=4 sw=4
Generated By: pequel Version 2.4-5, Build: Wednesday November 16 21:56:42 GMT 2005
              : http://sourceforge.net/projects/pequel/
    Script Name : chain_pequel_pt1.pql
    Created On : Wed Nov 16 14:00:39 2005
    Perl Version: /usr/bin/perl 5.6.1 on solaris
    For
        #--
   Options:
       input_file(sample.data) input data filename
       optimize(1) optimize generated code.
       doc_title(Pequel Chaining Part-1 Example Script) document title.
       doc_email(sample@youraddress.com) document email entry.
       doc_version(2.3) document version for pequel script.
                                                use strict;
      use constant _I_PRODUCT_CODE
                                   => int
                                            0;
      use constant _I_COST_PRICE
                                  => int
                                           1;
      use constant _I_DESCRIPTION
                                   => int
      use constant _I_SALES_CODE
                                  => int
      use constant _I_SALES_PRICE use constant _I_SALES_QTY
                                => ..
=> int
int
      use constant _I_SALES_DATE
                                  => int
      use constant _I_LOCATION
                                  => int
      use constant _I_SALES_TOTAL use constant _O_LOCATION
                                  => int
=> int
                                            1;
      use constant _O_PRODUCT_CODE use constant _O_SALES_TOTAL
                                   => int
                                  => int
       local \= \n'';
       local $,="|";
       print STDERR '[examples/chain_pequel_pt1.pql ' . localtime() . "] Init";
       use constant VERBOSE => int 10000;
       use constant LAST_ICELL => int 8;
      my @I_VAL;
      my @O_VAL;
      my $_inprecs=0;
      my $key__I_LOCATION;
      my $previous_key__I_LOCATION = undef;
      my $key__I_PRODUCT_CODE;
      my $previous_key__I_PRODUCT_CODE = undef;
       foreach my $f (1..3) { $O_VAL[$f] = undef; }
     Sort:LOCATION(asc:string) PRODUCT_CODE(asc:string)
      open(DATA, q{sort -t'|' -y -k 8,8 -k 1,1 examples/sample.data 2>/dev/null |});
open(STDOUT, '|-', q{sort -t'|' -y -k 1,1 2>/dev/null});
       print STDERR '[examples/chain_pequel_pt1.pql ' . localtime() . "] Start";
       use Benchmark;
       my $benchmark_start = new Benchmark;
       while (<DATA>)
       {
          ++$ inprecs;
          print STDERR '[examples/chain_pequel_pt1.pql ' . localtime() . "] $_inprecs records." if ($_inprec
s % VERBOSE == 0);
          chomp;
          @I_VAL = split("[|]", $_);
          $key__I_LOCATION = $I_VAL[_I_LOCATION];
          $key__I_PRODUCT_CODE = $I_VAL[_I_PRODUCT_CODE];
          if (!defined($previous_key__I_LOCATION) || !defined($previous_key__I_PRODUCT_CODE))
              $previous_key__I_LOCATION = $key__I_LOCATION;
              $previous_key__I_PRODUCT_CODE = $key__I_PRODUCT_CODE;
          elsif ($previous_key__I_LOCATION ne $key__I_LOCATION || $previous_key__I_PRODUCT_CODE ne $key__I_P
RODUCT_CODE)
              print STDOUT
                 $0_VAL[_O_LOCATION],
                 $O_VAL[_O_PRODUCT_CODE],
                 $0_VAL[_O_SALES_TOTAL]
              $previous_key__I_LOCATION = $key__I_LOCATION;
              $previous_key__I_PRODUCT_CODE = $key__I_PRODUCT_CODE;
              @O_VAL = undef;
```

```
}
           $O_VAL[_O_LOCATION] = $I_VAL[_I_LOCATION];
           $0_VAL[_O_PRODUCT_CODE] = $I_VAL[_I_PRODUCT_CODE];
           $I_VAL[_I_SALES_TOTAL] = $I_VAL[_I_SALES_QTY] * $I_VAL[_I_SALES_PRICE];
           $0_VAL[_O_SALES_TOTAL] += $I_VAL[_I_SALES_TOTAL] unless ($I_VAL[_I_SALES_TOTAL] eq '');
       print STDOUT
          $0_VAL[_O_LOCATION],
           $0_VAL[_O_PRODUCT_CODE],
           $0_VAL[_O_SALES_TOTAL]
       close(STDOUT);
       close(DATA);
       print STDERR '[examples/chain_pequel_pt1.pql ' . localtime() . "] $_inprecs records.";
       my $benchmark_end = new Benchmark;
       my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
       print STDERR '[examples/chain_pequel_pt1.pql ' . localtime() . "] Code statistics: @{[timestr($benchma
rk_timediff)]}";
}
{
   package p_copy_input_copy_record_nt;
   sub copy input copy record nt
    !/usr/bin/perl
                    vim: syntax=perl ts=4 sw=4
    Generated By: pequel Version 2.4-5, Build: Wednesday November 16 21:56:42 GMT 2005
               : http://sourceforge.net/projects/pequel/
    Script Name : copy_record_NT.pql
    Created On : Wed Nov 16 14:00:48 2005
    Perl Version: /usr/bin/perl 5.6.1 on solaris
                  Options:
       optimize(1) optimize generated code.
        hash(1) Generate in memory. Input data can be unsorted.
        doc_title(Copy Record Example Script) document title.
        doc_email(sample@youraddress.com) document email entry.
        doc_version(2.3) document version for pequel script.
use strict;
       use Fcntl ':flock';
                                    => int
       use constant \_I\_LOCATION
       use constant _I_PRODUCT_CODE
                                    => int
       use constant _I_SALES_TOTAL
                                    => int
       use constant _I_LOCATION_NAME use constant _O_LOCATION_NAME
                                    => int
       2;
       local \= \n'';
       local $,="|";
       print STDERR '[examples/copy_record_NT.pql ' . localtime() . "] Init";
       use constant VERBOSE => int 10000;
       use constant LAST_ICELL => int 3;
       my @I_VAL;
       my %O_VAL;
       my $key;
       my $_inprecs=0;
       print STDERR '[examples/copy_record_NT.pql ' . localtime() . "] Start";
       use Benchmark;
       my $benchmark_start = new Benchmark;
       while (<STDIN>)
           ++$_inprecs;
          print STDERR '[examples/copy_record_NT.pql ' . localtime() . "] $_inprecs records." if ($_inprecs
% VERBOSE == 0);
           chomp;
           @I_VAL = split("[|]", $_);
           $key = ( $I_VAL[_I_PRODUCT_CODE] );
           $I_VAL[_I_LOCATION_NAME] = 'Northern Territory';
           $0_VAL{$key}{_O_LOCATION_NAME} = $I_VAL[_I_LOCATION_NAME];
           $O_VAL{$key}{_O_PRODUCT_CODE} = $I_VAL[_I_PRODUCT_CODE];
           $O_VAL{$key}{_O_SALES_TOTAL} += $I_VAL[_I_SALES_TOTAL] unless ($I_VAL[_I_SALES_TOTAL] eq '');
       }
       foreach $key (sort keys %O_VAL)
           flock(STDOUT, LOCK EX);
```

```
print STDOUT
                           $O_VAL{$key}{_O_LOCATION_NAME},
                           $0_VAL{$key}{_O_PRODUCT_CODE},
$0_VAL{$key}{_O_SALES_TOTAL}
                    flock(STDOUT, LOCK UN);
              }
              close(STDIN);
              print STDERR '[examples/copy_record_NT.pql ' . localtime() . "] $_inprecs records.";
              my $benchmark end = new Benchmark;
             my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
              \texttt{print STDERR '[examples/copy\_record\_NT.pql ' . local time() . "] Code statistics: @\{[timestr(\$benchmark of the statistics] | (a) | (b) | (b) | (b) | (b) | (c) | (c
 _timediff)]}";
                                 }
}
{
      package p_copy_input_copy_record_vic;
      sub copy_input_copy_record_vic
        !/usr/bin/perl
#-+-+-
        vim: syntax=perl ts=4 sw=4
       Generated By: pequel Version 2.4-5, Build: Wednesday November 16 21:56:42 GMT 2005
                             : http://sourceforge.net/projects/pequel/
        Script Name : copy_record_VIC.pql
        Created On : Wed Nov 16 14:00:46 2005
        Perl Version: /usr/bin/perl 5.6.1 on solaris
        For
       +-----
#
        Options:
               optimize(1) optimize generated code.
               hash(1) Generate in memory. Input data can be unsorted.
               doc_title(Copy Record Example Script) document title.
               doc_email(sample@youraddress.com) document email entry.
              doc_version(2.3) document version for pequel script.
             use strict;
             use Fcntl ':flock';
             use constant \_I\_LOCATION
                                                                    => int
             use constant _I_PRODUCT_CODE => int
use constant _I_SALES_TOTAL => int
              use constant _I_SALES_TOTAL
              use constant _I_LOCATION_NAME
                                                                      => int
              use constant _O_LOCATION_NAME => int
             1;
              local $\="\n";
              local $,="|";
              print STDERR '[examples/copy_record_VIC.pql ' . localtime() . "] Init";
              use constant VERBOSE => int 10000;
              use constant LAST_ICELL => int 3;
              my @I_VAL;
             my %O_VAL;
             my $key;
              my $ inprecs=0;
             print STDERR '[examples/copy_record_VIC.pql ' . localtime() . "] Start";
              use Benchmark;
              my $benchmark_start = new Benchmark;
              while (<STDIN>)
                    print STDERR '[examples/copy_record_VIC.pql ' . localtime() . "] $_inprecs records." if ($_inprecs
 % VERBOSE == 0);
                    chomp;
                     @I_VAL = split("[|]", $_);
                     $key = ( $I_VAL[_I_PRODUCT_CODE] );
                     $I_VAL[_I_LOCATION_NAME] = 'Victoria';
                     $0_VAL{$key}{_O_LOCATION_NAME} = $I_VAL[_I_LOCATION_NAME];
                     $O_VAL{$key}{_O_PRODUCT_CODE} = $I_VAL[_I_PRODUCT_CODE];
                     $O_VAL{$key}{_O_SALES_TOTAL} += $I_VAL[_I_SALES_TOTAL] unless ($I_VAL[_I_SALES_TOTAL] eq '');
              foreach $key (sort keys %O_VAL)
                     flock(STDOUT, LOCK_EX);
                    print STDOUT
                           $O_VAL{$key}{_O_LOCATION_NAME},
                            O_VAL{skey}_{O_PRODUCT_CODE},
                            $0_VAL{$key}{_O_SALES_TOTAL}
                     flock(STDOUT, LOCK_UN);
```

```
}
      close(STDIN);
      print STDERR '[examples/copy_record_VIC.pql ' . localtime() . "] $_inprecs records.";
      my $benchmark end = new Benchmark;
      my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
      print STDERR '[examples/copy_record_VIC.pql ' . localtime() . "] Code statistics: @{[timestr($benchmar
k_timediff)]}";
#-----
  }
}
{
   package p copy input copy record nsw;
   sub copy_input_copy_record_nsw
    !/usr/bin/perl
vim: syntax=perl ts=4 sw=4
#-+
   Generated By: pequel Version 2.4-5, Build: Wednesday November 16 21:56:42 GMT 2005
              : http://sourceforge.net/projects/pequel/
    Script Name : copy_record_NSW.pql
    Created On : Wed Nov 16 14:00:44 2005
    Perl Version: /usr/bin/perl 5.6.1 on solaris
#-+
  Options:
      optimize(1) optimize generated code.
       hash(1) Generate in memory. Input data can be unsorted.
       doc_title(Copy Record Example Script) document title.
       doc_email(sample@youraddress.com) document email entry.
      doc_version(2.3) document version for pequel script.
                                                       #-----
      use strict;
      use Fcntl ':flock';
      use constant _I_LOCATION
                                  => int
                                => 1..
=> int
=> int
      use constant _I_PRODUCT_CODE
                                            1;
      use constant _I_SALES_TOTAL
      use constant _I_LOCATION_NAME => int
use constant _O_LOCATION_NAME => int
      local \= \n'';
      local $,="|";
      print STDERR '[examples/copy_record_NSW.pql ' . localtime() . "] Init";
      use constant VERBOSE => int 10000;
      use constant LAST_ICELL => int 3;
      my @I_VAL;
      my %O_VAL;
      my $key;
      my $_inprecs=0;
      print STDERR '[examples/copy_record_NSW.pql ' . localtime() . "] Start";
      use Benchmark;
      my $benchmark_start = new Benchmark;
      while (<STDIN>)
          ++$ inprecs;
          print STDERR '[examples/copy_record_NSW.pql ' . localtime() . "] $_inprecs records." if ($_inprecs
% VERBOSE == 0);
          chomp;
          @I_VAL = split("[|]", $_);
          $key = ( $I_VAL[_I_PRODUCT_CODE] );
          $I_VAL[_I_LOCATION_NAME] = 'New South Wales';
          $O_VAL{$key}{_O_LOCATION_NAME} = $I_VAL[_I_LOCATION_NAME];
          $O_VAL{$key}{_O_PRODUCT_CODE} = $I_VAL[_I_PRODUCT_CODE];
          $O_VAL{$key}{_O_SALES_TOTAL} += $I_VAL[_I_SALES_TOTAL] unless ($I_VAL[_I_SALES_TOTAL] eq '');
      }
      foreach $key (sort keys %O_VAL)
      {
          flock(STDOUT, LOCK_EX);
          print STDOUT
             O_VAL{skey}_{O_LOCATION_NAME},
             $0_VAL{$key}{_O_PRODUCT_CODE},
             $0_VAL{$key}{_0_SALES_TOTAL}
          flock(STDOUT, LOCK_UN);
      }
      close(STDIN);
      print STDERR '[examples/copy_record_NSW.pql ' . localtime() . "] $_inprecs records.";
      my $benchmark_end = new Benchmark;
      my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
```

7. ABOUT PEQUEL

This document was generated by Pequel.

https://sourceforge.net/projects/pequel/

COPYRIGHT

Copyright ©1999-2005, Mario Gaffiero. All Rights Reserved. 'Pequel' TM Copyright ©1999-2005, Mario Gaffiero. All Rights Reserved.

This program and all its component contents is copyrighted free software by Mario Gaffiero and is released under the GNU General Public License (GPL), Version 2, a copy of which may be found at http://www.opensource.org/licenses/gpl-license.html

Pequel is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Pequel is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with Pequel; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA