examples/chain_pequel_pt2.pql

by *Pequel*

sample@youraddress.com

Pequel Chaining Part-2 Example Script

2.3

Table of Contents
Pequel Chaining Part-2 Example Script

SCRIPT NAME

examples/chain_pequel_pt2.pql

DESCRIPTION

This example demonstrates Pequel script 'chaining'. By specifying a pequel script name for the 'input_file' option the input data stream will result by piping the output from executing the script specified in the 'input_file' option. Both scripts are executed simultaneously — with the input_file script as the child and this script as the parent. Beware of circular chaining! It is up to the user to ensure that this does not occur. The input-section should match the output field format of the input_file script. Additional derived input fields may be appended. The field names specified in the sort-by section must exist in the output for the input_file script. The sort-by section in the main script overrides the sort-output section in the sub script. Any 'output_file' option specification in the sub script is not allowed.

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 **grouped** by the input field ***LOCATION*** (*string*).

1.1 *LOCATION*

Output Field

Description
Set to input field ***LOCATION***

1.2 *COUNT_PRODUCT_CODE*

Output Field

Description
Distinct aggregation on input field ***PRODUCT_CODE***.

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/chain_pequel_pt2.pql

2.5 *input_file*
input data filename: chain_pequel_pt1.pql

2.6 *header*
write header record to output.: 0

2.7 *optimize*
optimize generated code.: 1

2.8 *doc_title*
document title.: Pequel Chaining Part-2 Example Script

2.9 *doc_email*
document email entry.: sample@youraddress.com

2.10 *doc_version*
document version for pequel script.: 2.3

3. TABLES

4. TABLE INFORMATION SUMMARY

4.1 Table List Sorted By Table Name

5. EXAMPLES/CHAIN_PEQUEL_PT2.PQL

options

```
prefix(examples)
pequeldoc(pdf)
detail(1)
script_name(examples/chain_pequel_pt2.pql)
input_file(chain_pequel_pt1.pql)
header(0)
optimize(1)
doc_title(Pequel Chaining Part-2 Example Script)
doc_email(sample@youraddress.com)
doc_version(2.3)
```

description

```
This example demonstrates Pequel script 'chaining'. By specifying a pequel script name for
the 'input_file' option
the input data stream will result by piping the output from executing
the script specified in the 'input_file' option.
Both scripts are executed simultaneously -- with the input_file script as the child and this
script as the parent. Beware of circular chaining! It is up to the user to ensure that this
does not occur.
The input-section should match the output field format of the input_file script. Additional
derived input fields may be appended.
The field names specified in the sort-by section must exist in the output for the input_file
script.
The sort-by section in the main script overrides the sort-output section in the sub script.
Any 'output_file' option specification in the sub script is not allowed.
```

input section

```
LOCATION
PRODUCT_CODE
SALES_TOTAL
```

group by

```
LOCATION string
```

output section

```
string    LOCATION           LOCATION
numeric   COUNT_PRODUCT_CODE distinct PRODUCT_CODE
decimal   SALES_TOTAL        sum SALES_TOTAL
```

sort output

```
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 : chain_pequel_pt2.pql
#Created On  : Wed Nov 16 13:55:37 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/chain_pequel_pt2.pql) script filename
#input_file(chain_pequel_pt1.pql) input data filename
#header(0) write header record to output.
#optimize(1) optimize generated code.
#doc_title(Pequel Chaining Part-2 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_LOCATION              => int    0;
use constant _I_PRODUCT_CODE          => int    1;
use constant _I_SALES_TOTAL           => int    2;
use constant _O_LOCATION              => int    1;
use constant _O_COUNT_PRODUCT_CODE    => int    2;
use constant _O_SALES_TOTAL           => int    3;
local $\="\n";
local $,="|";
print STDERR '[examples/chain_pequel_pt2.pql ' . localtime() . "] Init";
use constant VERBOSE => int 10000;
use constant LAST_ICELL => int 2;
my @I_VAL;
my @O_VAL;
my $_inprecs=0;
my %DISTINCT;
my $key__I_LOCATION;
my $previous_key__I_LOCATION = undef;
foreach my $f (1..3) { $O_VAL[$f] = undef; }
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 3nr,3nr 2>/dev/null});
print STDERR '[examples/chain_pequel_pt2.pql ' . localtime() . "] Start";
use Benchmark;
my $benchmark_start = new Benchmark;
while (<READ_CHAIN_PEQUEL_PT1>)
{
    ++$_inprecs;
    print STDERR '[examples/chain_pequel_pt2.pql ' . localtime() . "] $_inprecs records." if ($_inprecs % VERB
OSE == 0);
    chomp;
    @I_VAL = split("[|]", $_);
    $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)
    {
        print STDOUT
            $O_VAL[_O_LOCATION],
            $O_VAL[_O_COUNT_PRODUCT_CODE],
            $O_VAL[_O_SALES_TOTAL]
        ;
        $previous_key__I_LOCATION = $key__I_LOCATION;
        @O_VAL = undef;
        %DISTINCT = undef;
    }

    $O_VAL[_O_LOCATION] = $I_VAL[_I_LOCATION];
    $O_VAL[_O_COUNT_PRODUCT_CODE]++
        if (defined($I_VAL[_I_PRODUCT_CODE]) && ++$DISTINCT{_O_COUNT_PRODUCT_CODE}{qq{$I_VAL[_I_PRODUCT_CODE]}
```

```
} == 1);
    $O_VAL[_O_SALES_TOTAL] += $I_VAL[_I_SALES_TOTAL] unless ($I_VAL[_I_SALES_TOTAL] eq '');
}

print STDOUT
    $O_VAL[_O_LOCATION],
    $O_VAL[_O_COUNT_PRODUCT_CODE],
    $O_VAL[_O_SALES_TOTAL]
;
close(STDOUT);
close(READ_CHAIN_PEQUEL_PT1);
print STDERR '[examples/chain_pequel_pt2.pql ' . localtime() . "] $_inprecs records.";
my $benchmark_end = new Benchmark;
my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
print STDERR '[examples/chain_pequel_pt2.pql ' . localtime() . "] Code statistics: @{[timestr($benchmark_timed
iff)]}";
#-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
{
    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 13:55:37 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    2;
        use constant _I_SALES_CODE      => int    3;
        use constant _I_SALES_PRICE     => int    4;
        use constant _I_SALES_QTY       => int    5;
        use constant _I_SALES_DATE      => int    6;
        use constant _I_LOCATION        => int    7;
        use constant _I_SALES_TOTAL     => int    8;
        use constant _O_LOCATION        => int    1;
        use constant _O_PRODUCT_CODE    => int    2;
        use constant _O_SALES_TOTAL     => int    3;
        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
                    $O_VAL[_O_LOCATION],
                    $O_VAL[_O_PRODUCT_CODE],
                    $O_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];
            $O_VAL[_O_PRODUCT_CODE] = $I_VAL[_I_PRODUCT_CODE];
            $I_VAL[_I_SALES_TOTAL] = $I_VAL[_I_SALES_QTY] * $I_VAL[_I_SALES_PRICE];
            $O_VAL[_O_SALES_TOTAL] += $I_VAL[_I_SALES_TOTAL] unless ($I_VAL[_I_SALES_TOTAL] eq '');
        }

        print STDOUT
            $O_VAL[_O_LOCATION],
            $O_VAL[_O_PRODUCT_CODE],
            $O_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)]}";
#-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    }

}
```

7. ABOUT PEQUEL

This document was generated by Pequel.

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