



chain_pequel_pt1.pql

by *Pequel*

sample@youraddress.com

Pequel Chaining Part-1 Example Script

2.3

Table of Contents

Pequel Chaining Part-1 Example Script

SCRIPT NAME	1
DESCRIPTION	1
1. PROCESS DETAILS	1
1.1 LOCATION	1
Description	1
1.2 PRODUCT_CODE	1
Description	1
1.3 SALES_TOTAL	1
Description	1
2. CONFIGURATION SETTINGS	2
2.1 pequeldoc	2
2.2 detail	2
2.3 script_name	2
2.4 input_file	2
2.5 optimize	2
2.6 doc_title	2
2.7 doc_email	2
2.8 doc_version	2
3. TABLES	3
4. TABLE INFORMATION SUMMARY	4
4.1 Table List Sorted By Table Name	4
5. CHAIN_PEQUEL_PT1.PQL	5
options	5
description	5
sort by	5
group by	5
input section	5
output section	5
6. PEQUEL GENERATED PROGRAM	6
7. ABOUT PEQUEL	8
COPYRIGHT	8

SCRIPT NAME

chain_pequel_pt1.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 executing the specified script.

1. PROCESS DETAILS

Input records are read from sample.data. The input record contains **8** 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 stream is **sorted** by the input fields **LOCATION** (*string*) and **PRODUCT_CODE** (*string*).

Input records are **grouped** by the input fields **LOCATION** (*string*) and **PRODUCT_CODE** (*string*).

1.1 LOCATION

Output Field

DescriptionSet to input field **LOCATION****1.2 PRODUCT_CODE**

Output Field

DescriptionSet to input field **PRODUCT_CODE****1.3 SALES_TOTAL**

Output Field

Description**Sum** aggregation on input field **SALES_TOTAL**.

2. CONFIGURATION SETTINGS

2.1 *pequeldoc*

generate pod / pdf pequel script Reference Guide.: pdf

2.2 *detail*

Include Pequel Generated Program chapter in Pequeldoc: 1

2.3 *script_name*

script filename: chain_pequel_pt1.pql

2.4 *input_file*

input data filename: sample.data

2.5 *optimize*

optimize generated code.: 1

2.6 *doc_title*

document title.: Pequel Chaining Part-1 Example Script

2.7 *doc_email*

document email entry.: sample@youraddress.com

2.8 *doc_version*

document version for pequel script.: 2.3

3. TABLES

4. TABLE INFORMATION SUMMARY

4.1 Table List Sorted By Table Name

5. CHAIN_PEQUEL_PT1.PQL

options

```
pequeldoc(pdf)
detail(1)
script_name(chain_pequel_pt1.pql)
input_file(sample.data)
optimize(1)
doc_title(Pequel Chaining Part-1 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 executing the specified script.

sort by

```
LOCATION string
PRODUCT_CODE string
```

group by

```
LOCATION string
PRODUCT_CODE string
```

input section

```
PRODUCT_CODE
COST_PRICE
DESCRIPTION
SALES_CODE
SALES_PRICE
SALES_QTY
SALES_DATE
LOCATION
SALES_TOTAL => SALES_QTY * SALES_PRICE
```

output section

```
string    LOCATION    LOCATION
string    PRODUCT_CODE PRODUCT_CODE
decimal   SALES_TOTAL sum SALES_TOTAL
```

6. PEQUEL GENERATED PROGRAM

```
# vim: syntax=perl ts=4 sw=4
#-----
#Generated By: pequel Version 2.3-2, Build: Thursday September 22 19:56:03 BST 2005
#           : https://sourceforge.net/projects/pequel/
#Script Name : chain_pequel_pt1.pql
#Created On  : Tue Sep 27 14:46:33 2005
#For         :
#-----
#Options:
#pequeldoc(pdf) generate pod / pdf pequel script Reference Guide.
#detail(1) Include Pequel Generated Program chapter in Pequeldoc
#script_name(chain_pequel_pt1.pql) script filename
#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;
local $\\="\\n"; local $,="|";
print STDERR '[chain_pequel_pt1.pql ' . localtime() . "] Init";
use constant VERBOSE => int 10000;
use constant LAST_ICELL => int 8;
my @I_VAL;
my @O_VAL;
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; }
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;
open(DATA, q{sort -t'|' -y -k 8,8 -k 1,1 sample.data |});
print STDERR '[chain_pequel_pt1.pql ' . localtime() . "] Start";
use Benchmark;
my $benchmark_start = new Benchmark;
while (<DATA>)
{
    print STDERR '[chain_pequel_pt1.pql ' . localtime() . "] $. records." if ($. % 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_PRODUCT_C
ODE)
    {
        print
            $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
    $O_VAL[_O_LOCATION],
```

```
    $O_VAL[_O_PRODUCT_CODE],
    $O_VAL[_O_SALES_TOTAL]
;
print STDERR '[chain_pequel_pt1.pql ' . localtime() . "] $. records.";
my $benchmark_end = new Benchmark;
my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
print STDERR '[chain_pequel_pt1.pql ' . localtime() . "] Code statistics: @{{timestr($benchmark_timediff)}}";
#-+-+-+-----
```

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

