



# examples/chain\_pequel\_pt2.pql

by *Pequel*

---

[sample@youraddress.com](mailto:sample@youraddress.com)

## Pequel Chaining Part-2 Example Script

2.3



# Table of Contents

## Pequel Chaining Part-2 Example Script

SCRIPT NAME	1
DESCRIPTION	1
1. PROCESS DETAILS	1
1.1 LOCATION	1
Description	1
1.2 COUNT_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 prefix	2
2.4 script_name	2
2.5 input_file	2
2.6 header	2
2.7 optimize	2
2.8 doc_title	2
2.9 doc_email	2
2.10 doc_version	2
3. TABLES	3
4. TABLE INFORMATION SUMMARY	4
4.1 Table List Sorted By Table Name	4
5. EXAMPLES/CHAIN_PEQUEL_PT2.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	9
COPYRIGHT	9



## 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 executing the specified script. 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. Currently 'sort by' is not supported in the parent script.

## 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 stream is **sorted** by the input field **LOCATION** (*string*).

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 *pequeldoc*

generate pod / pdf pequel script Reference Guide.: pdf

### 2.2 *detail*

Include Pequel Generated Program chapter in Pequeldoc: 1

### 2.3 *prefix*

directory pathname prefix.: examples

### 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.: 1

### 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*

```
pequeldoc(pdf)
detail(1)
prefix(examples)
script_name(examples/chain_pequel_pt2.pql)
input_file(chain_pequel_pt1.pql)
header(1)
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 executing the specified script. 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. Currently 'sort by' is not supported in the parent script.

### *sort by*

```
LOCATION string
```

### *group by*

```
LOCATION string
```

### *input section*

```
LOCATION
PRODUCT_CODE
SALES_TOTAL
```

### *output section*

string	LOCATION	LOCATION
numeric	COUNT_PRODUCT_CODE	distinct PRODUCT_CODE
decimal	SALES_TOTAL	sum SALES_TOTAL

## 6. PEQUEL GENERATED PROGRAM

```
# vim: syntax=perl ts=4 sw=4
#-----
#Generated By: pequel Version 2.3-3, Build: Monday October 3 23:16:49 BST 2005
#           : https://sourceforge.net/projects/pequel/
#Script Name : examples/chain_pequel_pt2.pql
#Created On  : Tue Oct 4 10:35:09 2005
#For         :
#-----
#Options:
#pequeldoc(pdf) generate pod / pdf pequel script Reference Guide.
#detail(1) Include Pequel Generated Program chapter in Pequeldoc
#prefix(examples) directory pathname prefix.
#script_name(examples/chain_pequel_pt2.pql) script filename
#input_file(chain_pequel_pt1.pql) input data filename
#header(1) 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;
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 %DISTINCT;
my $key__I_LOCATION;
my $previous_key__I_LOCATION = undef;
foreach my $f (1..3) { $O_VAL[$f] = undef; }
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;
my $pid = open(CHAIN_PEQUEL_PT1, '-|'); # Fork
my $count=0;
if (!$pid) # Child
{
    &p_execPequelCHAIN_PEQUEL_PT1::execPequelCHAIN_PEQUEL_PT1;
    exit(0);
}

open(STDOUT, '|-', q{sort -t'|' -y -k 3n,3n |});
&PrintHeader();
print STDERR "[examples/chain_pequel_pt2.pql ' . localtime() . "] Start";
use Benchmark;
my $benchmark_start = new Benchmark;
while (<CHAIN_PEQUEL_PT1)
{
    print STDERR "[examples/chain_pequel_pt2.pql ' . localtime() . "] $. records." if ($. % VERBOSE == 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
            $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
```

```

    $O_VAL[_O_LOCATION],
    $O_VAL[_O_COUNT_PRODUCT_CODE],
    $O_VAL[_O_SALES_TOTAL]
;
close(STDOUT);
print STDERR '[examples/chain_pequel_pt2.pql ' . localtime() . "] $. 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_timediff)])";
#-----
sub PrintHeader
{
    local $\="\n";
    local $,="|";
    print
        'LOCATION',
        'COUNT_PRODUCT_CODE',
        'SALES_TOTAL'
    ;
}

{
    package p_execPequelCHAIN_PEQUEL_PT1;
    sub execPequelCHAIN_PEQUEL_PT1
    {
        # vim: syntax=perl ts=4 sw=4
        #-----
        #   Generated By: pequel Version 2.3-3, Build: Monday October  3 23:16:49 BST 2005
        #           : https://sourceforge.net/projects/pequel/
        #   Script Name : examples/chain_pequel_pt1.pql
        #   Created On  : Tue Oct  4 10:35:08 2005
        #   For         :
        #-----
        #   Options:
        #       input_file(sample.data) input data filename
        #       optimize(1) optimize generated code.
        #       hash(1) Generate in memory. Input data can be unsorted.
        #       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 '[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 $key;
        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{examples/sample.data}) || die "Cannot open examples/sample.data: $!";
        open(STDOUT, '|-', q{sort -t'|' -y -k 1,1 |});
        print STDERR '[examples/chain_pequel_pt1.pql ' . localtime() . "] Start";
        use Benchmark;
        my $benchmark_start = new Benchmark;
        while (<DATA>)
        {
            print STDERR '[examples/chain_pequel_pt1.pql ' . localtime() . "] $. records." if ($. % VERBOSE ==
0);
            chomp;
            @I_VAL = split("[|]", $_);
            $key = ( $I_VAL[_I_LOCATION] ) . '|' . ( $I_VAL[_I_PRODUCT_CODE] );
            $O_VAL{$key}{_O_LOCATION} = $I_VAL[_I_LOCATION];
            $O_VAL{$key}{_O_PRODUCT_CODE} = $I_VAL[_I_PRODUCT_CODE];
            $I_VAL[_I_SALES_TOTAL] = $I_VAL[_I_SALES_QTY] * $I_VAL[_I_SALES_PRICE];
            $O_VAL{$key}{_O_SALES_TOTAL} += $I_VAL[_I_SALES_TOTAL] unless ($I_VAL[_I_SALES_TOTAL] eq '');
        }

        foreach $key (sort keys %O_VAL)
        {
            print
                $O_VAL{$key}{_O_LOCATION},
                $O_VAL{$key}{_O_PRODUCT_CODE},

```

```

        $O_VAL{$key}{_O_SALES_TOTAL}
    ;
}

close(STDOUT);
print STDERR '[examples/chain_pequel_pt1.pql ' . localtime() . "] $. 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/>***

## 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

