



top_prod_by_loc.pql by *Pequel*

sample@youraddress.com

Pequel Table Example Script

2.3

Table of Contents

Pequel Table Example Script

SCRIPT NAME	1
DESCRIPTION	1
1. PROCESS DETAILS	1
1.1 LOCATION	1
Description	1
1.3 PRODUCT_CODE	1
Description	1
Aggregation condition	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 header	2
2.6 optimize	2
2.7 hash	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. TOP_PROD_BY_LOC.PQL	5
options	5
description	5
group by	5
input section	5
output section	5
6. PEQUEL GENERATED PROGRAM	6
7. ABOUT PEQUEL	8
COPYRIGHT	8

SCRIPT NAME

top_prod_by_loc.pql

DESCRIPTION

This script demonstrates the use of pequel tables. This script will be called by another Pequel script to load the table data via the 'load pequel table' section. The important thing here is to specify the 'input_file' option.

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

1.1 LOCATION

Output Field

Description

Set to input field **LOCATION**

1.3 PRODUCT_CODE

Output Field

Description

First aggregation on input field **PRODUCT_CODE**.

Aggregation condition

`sprintf("%.2f",SALES_TOTAL) eq sprintf("%.2f",_MAXSALES);`

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: top_prod_by_loc.pql

2.4 *input_file*

input data filename: sample.data

2.5 *header*

write header record to output.: 1

2.6 *optimize*

optimize generated code.: 1

2.7 *hash*

Generate in memory. Input data can be unsorted.: 1

2.8 *doc_title*

document title.: Pequel Table 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. TOP_PROD_BY_LOC.PQL

options

```
pequeldoc(pdf)
detail(1)
script_name(top_prod_by_loc.pql)
input_file(sample.data)
header(1)
optimize(1)
hash(1)
doc_title(Pequel Table Example Script)
doc_email(sample@youraddress.com)
doc_version(2.3)
```

description

This script demonstrates the use of pequel tables. This script will be called by another Pequel script to load the table data via the 'load pequel table' section. The important thing here is to specify the 'input_file' option.

group by

```
LOCATION 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
decimal   _MAXSALES   max SALES_TOTAL
string     PRODUCT_CODE first PRODUCT_CODE where sprintf("%.2f",SALES_TOTAL) eq sprintf("%.2f",_MAXSALES)
```

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 : top_prod_by_loc.pql
#Created On  : Tue Sep 27 14:45:43 2005
#For         :
#-----
#Options:
#pequeldoc(pdf) generate pod / pdf pequel script Reference Guide.
#detail(1) Include Pequel Generated Program chapter in Pequeldoc
#script_name(top_prod_by_loc.pql) script filename
#input_file(sample.data) input data filename
#header(1) write header record to output.
#optimize(1) optimize generated code.
#hash(1) Generate in memory. Input data can be unsorted.
#doc_title(Pequel Table 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 "[top_prod_by_loc.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_MAXSALES       => int    2;
use constant _O_PRODUCT_CODE   => int    3;
open(DATA, q{sample.data}) || die "Cannot open sample.data: $!";
&PrintHeader();
print STDERR "[top_prod_by_loc.pql ' . localtime() . "] Start";
use Benchmark;
my $benchmark_start = new Benchmark;
while (<DATA>)
{
    print STDERR "[top_prod_by_loc.pql ' . localtime() . "] $. records." if ($. % VERBOSE == 0);
    chomp;
    @I_VAL = split("[|]", $_);
    $key = ( $I_VAL[_I_LOCATION] );
    $O_VAL{$key}{_O_LOCATION} = $I_VAL[_I_LOCATION];
    $I_VAL[_I_SALES_TOTAL] = $I_VAL[_I_SALES_QTY] * $I_VAL[_I_SALES_PRICE];
    $O_VAL{$key}{_O_MAXSALES} = $I_VAL[_I_SALES_TOTAL]
        if (!defined($O_VAL{$key}{_O_MAXSALES}) || $I_VAL[_I_SALES_TOTAL] > $O_VAL{$key}{_O_MAXSALES});

    if (sprintf("%.2f", $I_VAL[_I_SALES_TOTAL]) eq sprintf("%.2f", $O_VAL{$key}{_O_MAXSALES})) {
        $O_VAL{$key}{_O_PRODUCT_CODE} = $I_VAL[_I_PRODUCT_CODE] if (!defined($O_VAL{$key}{_O_PRODUCT_CODE}));
    }
}

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

print STDERR "[top_prod_by_loc.pql ' . localtime() . "] $. records.";
my $benchmark_end = new Benchmark;
my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
print STDERR "[top_prod_by_loc.pql ' . localtime() . "] Code statistics: @{{timestr($benchmark_timediff)}}";
#-----
sub PrintHeader
{
    local $\\="\\n";
    local $,="|";
    print
        'LOCATION',

```

```
        'PRODUCT_CODE'  
        ;  
    }
```

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

