sales_ttl_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
1.2 SALES_TOTAL 1
Description 1
1.3 TOP_PRODUCT 1
Description 1
Derived Input Field Evaluation 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
3.1 TTOPPRODBYLOC 3
4. TABLE INFORMATION SUMMARY 4
4.1 Table List Sorted By Table Name 4
5. SALES_TTL_BY_LOC.PQL 5
options 5
description 5
load table 5
group by 5
input section 5
output section 5
6. PEQUEL GENERATED PROGRAM 6
7. ABOUT PEQUEL 9
COPYRIGHT 9

SCRIPT NAME

sales_ttl_by_loc.pql

DESCRIPTION

This script demonstrates the use of pequel tables. This scipt 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***

1.2 *SALES_TOTAL*

Output Field

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

1.3 *TOP_PRODUCT*

Output Field

Description
Set to input field ***TOP_PRODUCT***

Derived Input Field Evaluation

```
=> %TTOPPRODBYLOC(LOCATION)->PRODUCT_CODE
```

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: sales_ttl_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

3.1 *TTOPPRODBYLOC*

Table Type: ***external***
Data Source Filename: ***top_prod_by_loc.pql***
Key Field Number: ***1***

3.1.1 *PRODUCT_CODE = 2*

4. TABLE INFORMATION SUMMARY

4.1 Table List Sorted By Table Name

TTOPPRODBYLOC — ***1*** (*external*)

5. SALES_TTL_BY_LOC.PQL

options

```
pequeldoc(pdf)
detail(1)
script_name(sales_ttl_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 scipt 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.
```

load table

```
TTOPPRODBYLOC /* Table Name */ \
    top_prod_by_loc.pql /* Data Source Filename */ \
    1 /* Key Column Number */ \
     \
    PRODUCT_CODE = 2
```

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

TOP_PRODUCT => %TTOPPRODBYLOC(LOCATION)->PRODUCT_CODE
```

output section

```
string    LOCATION    LOCATION
decimal   SALES_TOTAL sum SALES_TOTAL
string    TOP_PRODUCT TOP_PRODUCT
```

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 : sales_ttl_by_loc.pql
#Created On  : Tue Sep 27 14:44:49 2005
#For         :
#-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
#Options:
#pequeldoc(pdf) generate pod / pdf pequel script Reference Guide.
#detail(1) Include Pequel Generated Program chapter in Pequeldoc
#script_name(sales_ttl_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 '[sales_ttl_by_loc.pql ' . localtime() . "] Init";
use constant VERBOSE => int 10000;
use constant LAST_ICELL => int 9;
my @I_VAL;
my %O_VAL;
my $key;
my $_TABLE_TTOPPRODBYLOC = &LoadTableTTOPPRODBYLOC; # ref to %$TTOPPRODBYLOC hash
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 _I_TOP_PRODUCT     => int    9;
use constant _O_LOCATION        => int    1;
use constant _O_SALES_TOTAL     => int    2;
use constant _O_TOP_PRODUCT     => int    3;
use constant _T_TTOPPRODBYLOC_FLD_PRODUCT_CODE   => int    0;
use constant _I_TTOPPRODBYLOC_LOCATION_FLD_KEY           => int   10;
use constant _I_TTOPPRODBYLOC_LOCATION_FLD_PRODUCT_CODE  => int   11;
open(DATA, q{sample.data})|| die "Cannot open sample.data: $!";
&PrintHeader();
print STDERR '[sales_ttl_by_loc.pql ' . localtime() . "] Start";
use Benchmark;
my $benchmark_start = new Benchmark;
while (<DATA>)
{
    print STDERR '[sales_ttl_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_SALES_TOTAL} += $I_VAL[_I_SALES_TOTAL] unless ($I_VAL[_I_SALES_TOTAL] eq '');
    $I_VAL[_I_TOP_PRODUCT] = $$_TABLE_TTOPPRODBYLOC{qq{$I_VAL[_I_LOCATION]}};
    $O_VAL{$key}{_O_TOP_PRODUCT} = $I_VAL[_I_TOP_PRODUCT];
}

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

print STDERR '[sales_ttl_by_loc.pql ' . localtime() . "] $. records.";
my $benchmark_end = new Benchmark;
my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
print STDERR '[sales_ttl_by_loc.pql ' . localtime() . "] Code statistics: @{[timestr($benchmark_timediff)]}";
#-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
#++++++ Table TTOPPRODBYLOC --> Type :Pequel::Type::Table::External::Pequel ++++++
sub LoadTableTTOPPRODBYLOC
{
    my %_TABLE_TTOPPRODBYLOC;
    print STDERR '[sales_ttl_by_loc.pql ' . localtime() . "] Loading table TTOPPRODBYLOC from top_prod_by_loc.
pql...";
    my $pid = open(TTOPPRODBYLOC, '-|'); # Fork
    my $count=0;
    if ($pid) # Parent
    {
        while (<TTOPPRODBYLOC>)
        {
            chomp;
            my (@flds) = split("[|]", $_, -1);
            $_TABLE_TTOPPRODBYLOC{$flds[0]} = $flds[ 1 ];
            print STDERR '[sales_ttl_by_loc.pql ' . localtime() . "] Table TTOPPRODBYLOC $. records..." if ($.
 % 100000 == 0);
        }

        $count=$.;
        close(TTOPPRODBYLOC);
    }

    else # Child
    {
        &p_LoadTableTTOPPRODBYLOC::LoadTableTTOPPRODBYLOC;
        exit(0);
    }

    print STDERR '[sales_ttl_by_loc.pql ' . localtime() . "] Table TTOPPRODBYLOC loaded $count records.";
    close(TTOPPRODBYLOC);
    return \%_TABLE_TTOPPRODBYLOC;
}

{
    package p_LoadTableTTOPPRODBYLOC;
    sub LoadTableTTOPPRODBYLOC
    {
#     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 14:44:48 2005
#    For         :
#-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
#    Options:
#        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(2.2) 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_C
ODE}));
            }
        }

        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_timedif
f)]}";
#-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        sub PrintHeader
        {
            local $\="\n";
            local $,="|";
            print
                ’LOCATION’,
                ’PRODUCT_CODE’
                ;
        }

    }

}

sub PrintHeader
{
    local $\="\n";
    local $,="|";
    print
        ’LOCATION’,
        ’SALES_TOTAL’,
        ’TOP_PRODUCT’
        ;
}
```

7. ABOUT PEQUEL

This document was generated by Pequel.

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

COPYRIGHT