hash_option.pql

by *Pequel*

sample@youraddress.com

Hash Option Example Script

2.2

Table of Contents
Hash Option Example Script

SCRIPT NAME

hash_option.pql

DESCRIPTION

This example demonstrates the use of the 'hash' option. With the 'hash' option input data sorting is not required — the data will be aggregated in memory. For this reason the 'hash' option should only be used when the total number of groups is small depending on the amount of available memory.

1. PROCESS DETAILS

Input records are read from standard input. The input record contains ***8*** fields. Fields are delimited by the '|' character.

Output records are written to standard output. The output record contains ***9*** 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 *MIN_COST_PRICE*

Output Field

Description
Min aggregation on input field ***COST_PRICE***.

1.3 *MAX_COST_PRICE*

Output Field

Description
Max aggregation on input field ***COST_PRICE***.

1.5 *SALES_CODE_1*

Output Field

Description
First aggregation on input field ***SALES_CODE***.

Aggregation condition
_DISTINCT_SALES_CODE == 1;

1.6 *SALES_CODE_2*

Output Field

Description
First aggregation on input field ***SALES_CODE***.

Aggregation condition

_DISTINCT_SALES_CODE == 2;

1.7 *SALES_CODE_3*

Output Field

Description
First aggregation on input field ***SALES_CODE***.

Aggregation condition
_DISTINCT_SALES_CODE == 3;

1.8 *SALES_CODE_4*

Output Field

Description
First aggregation on input field ***SALES_CODE***.

Aggregation condition
_DISTINCT_SALES_CODE == 4;

1.9 *SALES_CODE_5*

Output Field

Description
First aggregation on input field ***SALES_CODE***.

Aggregation condition
_DISTINCT_SALES_CODE == 5;

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: hash_option.pql

2.4 *header*
write header record to output.: 1

2.5 *optimize*
optimize generated code.: 1

2.6 *hash*
Generate in memory. Input data can be unsorted.: 1

2.7 *doc_title*
document title.: Hash Option Example Script

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

2.9 *doc_version*
document version for pequel script.: 2.2

3. TABLES

4. TABLE INFORMATION SUMMARY

4.1 Table List Sorted By Table Name

5. HASH_OPTION.PQL

options

```
pequeldoc(pdf)
detail(1)
script_name(hash_option.pql)
header(1)
optimize(1)
hash(1)
doc_title(Hash Option Example Script)
doc_email(sample@youraddress.com)
doc_version(2.2)
```

description

```
This example demonstrates the use of the 'hash' option. With the 'hash' option
input data sorting is not required -- the data will be aggregated in memory.
For this reason the 'hash' option should only be used when the total number of
groups is small
depending on the amount of available memory.
```

group by

```
LOCATION string
```

input section

```
PRODUCT_CODE
COST_PRICE
DESCRIPTION
SALES_CODE
SALES_PRICE
SALES_QTY
SALES_DATE
LOCATION
```

output section

```
string    LOCATION             LOCATION
numeric   MIN_COST_PRICE       min COST_PRICE
numeric   MAX_COST_PRICE       max COST_PRICE
numeric   _DISTINCT_SALES_CODE distinct SALES_CODE
string    SALES_CODE_1         first SALES_CODE where _DISTINCT_SALES_CODE == 1
string    SALES_CODE_2         first SALES_CODE where _DISTINCT_SALES_CODE == 2
string    SALES_CODE_3         first SALES_CODE where _DISTINCT_SALES_CODE == 3
string    SALES_CODE_4         first SALES_CODE where _DISTINCT_SALES_CODE == 4
string    SALES_CODE_5         first SALES_CODE where _DISTINCT_SALES_CODE == 5
```

6. PEQUEL GENERATED PROGRAM

```
# vim: syntax=perl ts=4 sw=4
#-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
#Generated By: pequel Version 2.2-9, Build: Tuesday September 13 08:43:08 BST 2005
#            : https://sourceforge.net/projects/pequel/
#Script Name : hash_option.pql
#Created On  : Tue Sep 13 10:29:33 2005
#For         :
#-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
#Options:
#pequeldoc(pdf) generate pod / pdf pequel script Reference Guide.
#detail(1) Include Pequel Generated Program chapter in Pequeldoc
#script_name(hash_option.pql) script 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(Hash Option 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 '[hash_option.pql ' . localtime() . "] Init";
use constant VERBOSE => int 10000;
use constant LAST_ICELL => int 7;
my @I_VAL;
my %O_VAL;
my $key;
my %DISTINCT;
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 _O_LOCATION                => int    1;
use constant _O_MIN_COST_PRICE          => int    2;
use constant _O_MAX_COST_PRICE          => int    3;
use constant _O__DISTINCT_SALES_CODE    => int    4;
use constant _O_SALES_CODE_1            => int    5;
use constant _O_SALES_CODE_2            => int    6;
use constant _O_SALES_CODE_3            => int    7;
use constant _O_SALES_CODE_4            => int    8;
use constant _O_SALES_CODE_5            => int    9;
&PrintHeader();
print STDERR '[hash_option.pql ' . localtime() . "] Start";
use Benchmark;
my $benchmark_start = new Benchmark;
while (<STDIN>)
{
    print STDERR '[hash_option.pql ' . localtime() . "] $. records." if ($. % VERBOSE == 0);
    chomp;
    @I_VAL = split("[|]", $_);
    $key = ( $I_VAL[_I_LOCATION] );
    $O_VAL{$key}{_O_LOCATION} = $I_VAL[_I_LOCATION];
    $O_VAL{$key}{_O_MIN_COST_PRICE} = $I_VAL[_I_COST_PRICE]
        if (!defined($O_VAL{$key}{_O_MIN_COST_PRICE}) || $I_VAL[_I_COST_PRICE] < $O_VAL{$key}{_O_MIN_COST_PRIC
E});
    $O_VAL{$key}{_O_MAX_COST_PRICE} = $I_VAL[_I_COST_PRICE]
        if (!defined($O_VAL{$key}{_O_MAX_COST_PRICE}) || $I_VAL[_I_COST_PRICE] > $O_VAL{$key}{_O_MAX_COST_PRIC
E});
    $O_VAL{$key}{_O__DISTINCT_SALES_CODE}++ if (defined($I_VAL[_I_SALES_CODE]) && ++$DISTINCT{$key}{_O__DISTIN
CT_SALES_CODE}{qq{$I_VAL[_I_SALES_CODE]}} == 1);

    if ($O_VAL{$key}{_O__DISTINCT_SALES_CODE} == 1) {
        $O_VAL{$key}{_O_SALES_CODE_1} = $I_VAL[_I_SALES_CODE] if (!defined($O_VAL{$key}{_O_SALES_CODE_1}));
    }
    elsif ($O_VAL{$key}{_O__DISTINCT_SALES_CODE} == 2) {
        $O_VAL{$key}{_O_SALES_CODE_2} = $I_VAL[_I_SALES_CODE] if (!defined($O_VAL{$key}{_O_SALES_CODE_2}));
    }
    elsif ($O_VAL{$key}{_O__DISTINCT_SALES_CODE} == 3) {
        $O_VAL{$key}{_O_SALES_CODE_3} = $I_VAL[_I_SALES_CODE] if (!defined($O_VAL{$key}{_O_SALES_CODE_3}));
    }
    elsif ($O_VAL{$key}{_O__DISTINCT_SALES_CODE} == 4) {
        $O_VAL{$key}{_O_SALES_CODE_4} = $I_VAL[_I_SALES_CODE] if (!defined($O_VAL{$key}{_O_SALES_CODE_4}));
    }
    elsif ($O_VAL{$key}{_O__DISTINCT_SALES_CODE} == 5) {
        $O_VAL{$key}{_O_SALES_CODE_5} = $I_VAL[_I_SALES_CODE] if (!defined($O_VAL{$key}{_O_SALES_CODE_5}));
    }
```

```
}

foreach $key (sort  keys %O_VAL)
{
    print
        $O_VAL{$key}{_O_LOCATION},
        $O_VAL{$key}{_O_MIN_COST_PRICE},
        $O_VAL{$key}{_O_MAX_COST_PRICE},
        $O_VAL{$key}{_O_SALES_CODE_1},
        $O_VAL{$key}{_O_SALES_CODE_2},
        $O_VAL{$key}{_O_SALES_CODE_3},
        $O_VAL{$key}{_O_SALES_CODE_4},
        $O_VAL{$key}{_O_SALES_CODE_5}
    ;
}

print STDERR '[hash_option.pql ' . localtime() . "] $. records.";
my $benchmark_end = new Benchmark;
my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
print STDERR '[hash_option.pql ' . localtime() . "] Code statistics: @{[timestr($benchmark_timediff)]}";
#-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
sub PrintHeader
{
    local $\="\n";
    local $,="|";
    print
        'LOCATION',
        'MIN_COST_PRICE',
        'MAX_COST_PRICE',
        'SALES_CODE_1',
        'SALES_CODE_2',
        'SALES_CODE_3',
        'SALES_CODE_4',
        'SALES_CODE_5'
        ;
}
```

7. ABOUT PEQUEL

This document was generated by Pequel.

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