group_by_derived_2.pql

by *Pequel*

sample@youraddress.com

Group By Derived Example Script

2.2

Table of Contents
Group By Derived Example Script

SCRIPT NAME

group_by_derived_2.pql

DESCRIPTION

This example demonstrates the use of a derived (calculated) field as the grouping field. In this example it is assumed that the input data contains mixed case values for LOCATION. The 'hash' option is important here because grouping is based on exact values — that is LOCATION's 'NSW' and 'Nsw' are not equal but converting both to upper case make them equal. With the 'hash' option the input data need not be sorted because the output is generated in memory using Perl's associative arrays. 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 ***3*** fields. Fields are delimited by the '|' character.

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

1.1 *FIXED_LOC_CODE*

Output Field

Description

Set to input field ***FIXED_LOC_CODE***

Derived Input Field Evaluation

```
=> %TCITY(LOCATION)->2 || LOCATION
```

1.2 *STATE_NAME*

Output Field

Description

Set to input field ***STATE_NAME***

Derived Input Field Evaluation

```
=> %TSTATE(FIXED_LOC_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: group_by_derived_2.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.: Group By Derived 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

3.1 *TCITY*

Table Type: ***local***

Data

SYD — Sydney NSW
MEL — Melbourne VIC
PER — Perth WA
ALIC — Alice Springs NT

3.2 *TSTATE*

Table Type: ***local***

Data

WA — Western Australia
NSW — New South Wales
SA — South Australia
QLD — Queensland
NT — Northern Territory
VIC — Victoria

4. TABLE INFORMATION SUMMARY

4.1 Table List Sorted By Table Name

TCITY — ***1*** (*local*)
TSTATE — ***2*** (*local*)

5. GROUP_BY_DERIVED_2.PQL

options

```
pequeldoc(pdf)
detail(1)
script_name(group_by_derived_2.pql)
header(1)
optimize(1)
hash(1)
doc_title(Group By Derived Example Script)
doc_email(sample@youraddress.com)
doc_version(2.2)
```

description

```
This example demonstrates the use of a derived (calculated)
field as the grouping field. In this example it is assumed that
the input data contains mixed case values for LOCATION.
The 'hash' option is important here because grouping is based
on exact values -- that is
LOCATION's 'NSW' and 'Nsw' are not
equal
but converting both to upper case make them equal. With
the 'hash' option
the input data need not be sorted because
the output is generated in memory using Perl's associative arrays.
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.
```

init table

```
TCITY SYD Sydney NSW
TCITY MEL Melbourne VIC
TCITY PER Perth WA
TCITY ALIC Alice Springs NT

TSTATE WA Western Australia
TSTATE NSW New South Wales
TSTATE SA South Australia
TSTATE QLD Queensland
TSTATE NT Northern Territory
TSTATE VIC Victoria
```

group by

```
FIXED_LOC_CODE string
```

input section

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

FIXED_LOC_CODE => %TCITY(LOCATION)->2 || LOCATION

STATE_NAME => %TSTATE(FIXED_LOC_CODE)
```

output section

```
string    FIXED_LOC_CODE FIXED_LOC_CODE
string    STATE_NAME     STATE_NAME
decimal   SALES_TOTAL    sum SALES_TOTAL
```

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 : group_by_derived_2.pql
#Created On  : Tue Sep 13 10:28:59 2005
#For         :
#-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
#Options:
#pequeldoc(pdf) generate pod / pdf pequel script Reference Guide.
#detail(1) Include Pequel Generated Program chapter in Pequeldoc
#script_name(group_by_derived_2.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(Group By Derived 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 '[group_by_derived_2.pql ' . localtime() . "] Init";
use constant VERBOSE => int 10000;
use constant LAST_ICELL => int 10;
my @I_VAL;
my %O_VAL;
my $key;
my $_TABLE_TCITY = &InitLookupTCITY; # ref to %$TCITY hash
my $_TABLE_TSTATE = &InitLookupTSTATE; # ref to %$TSTATE 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_FIXED_LOC_CODE    => int    9;
use constant _I_STATE_NAME        => int   10;
use constant _O_FIXED_LOC_CODE    => int    1;
use constant _O_STATE_NAME        => int    2;
use constant _O_SALES_TOTAL       => int    3;
use constant _T_TCITY_FLD_1    => int    0;
use constant _T_TCITY_FLD_2    => int    1;
use constant _T_TSTATE_FLD_1   => int    0;
use constant _I_TCITY_LOCATION_FLD_KEY       => int   11;
use constant _I_TCITY_LOCATION_FLD_1         => int   12;
use constant _I_TCITY_LOCATION_FLD_2         => int   13;
use constant _I_TSTATE_FIXED_LOC_CODE_FLD_KEY => int   14;
use constant _I_TSTATE_FIXED_LOC_CODE_FLD_1  => int   15;
&PrintHeader();
print STDERR '[group_by_derived_2.pql ' . localtime() . "] Start";
use Benchmark;
my $benchmark_start = new Benchmark;
while (<STDIN>)
{
    print STDERR '[group_by_derived_2.pql ' . localtime() . "] $. records." if ($. % VERBOSE == 0);
    chomp;
    @I_VAL = split("[|]", $_);
    $I_VAL[_I_FIXED_LOC_CODE] = ${$$_TABLE_TCITY{qq{$I_VAL[_I_LOCATION]}}}[_T_TCITY_FLD_2] || $I_VAL[_I_LOCATI
ON];
    $key = ( $I_VAL[_I_FIXED_LOC_CODE] );
    $O_VAL{$key}{_O_FIXED_LOC_CODE} = $I_VAL[_I_FIXED_LOC_CODE];
    $I_VAL[_I_STATE_NAME] = $$_TABLE_TSTATE{qq{$I_VAL[_I_FIXED_LOC_CODE]}};
    $O_VAL{$key}{_O_STATE_NAME} = $I_VAL[_I_STATE_NAME];
    $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_FIXED_LOC_CODE},
        $O_VAL{$key}{_O_STATE_NAME},
        $O_VAL{$key}{_O_SALES_TOTAL}
    ;
}

print STDERR '[group_by_derived_2.pql ' . localtime() . "] $. records.";
```

```
my $benchmark_end = new Benchmark;
my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
print STDERR '[group_by_derived_2.pql ' . localtime() . "] Code statistics: @{[timestr($benchmark_timediff)]}"
;
#-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
#++++++ Table TCITY --> Type :Pequel::Type::Table::Local ++++++
sub InitLookupTCITY
{
    my %_TABLE_TCITY;
    %_TABLE_TCITY =
    (
        'ALIC' => ['Alice Springs', 'NT'],
        'MEL' => ['Melbourne', 'VIC'],
        'PER' => ['Perth', 'WA'],
        'SYD' => ['Sydney', 'NSW']
    );
    return \%_TABLE_TCITY;
}

#++++++ Table TSTATE --> Type :Pequel::Type::Table::Local ++++++
sub InitLookupTSTATE
{
    my %_TABLE_TSTATE;
    %_TABLE_TSTATE =
    (
        'NSW' => 'New South Wales',
        'NT' => 'Northern Territory',
        'QLD' => 'Queensland',
        'SA' => 'South Australia',
        'VIC' => 'Victoria',
        'WA' => 'Western Australia'
    );
    return \%_TABLE_TSTATE;
}

sub PrintHeader
{
    local $\="\n";
    local $,="|";
    print
        'FIXED_LOC_CODE',
        'STATE_NAME',
        'SALES_TOTAL'
        ;
}
```

7. ABOUT PEQUEL

This document was generated by Pequel.

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