examples/group_by_derived.pql

by *Pequel*

sample@youraddress.com

Group By Derived Example Script

2.2

Table of Contents
Group By Derived Example Script

SCRIPT NAME 1
DESCRIPTION 1
1. PROCESS DETAILS 1
1.1 FIXED_LOC_CODE 1
Description 1
Derived Input Field Evaluation 1
1.2 SALES_TOTAL 1
Description 1
2. CONFIGURATION SETTINGS 2
2.1 prefix 2
2.2 pequeldoc 2
2.3 detail 2
2.4 script_name 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. EXAMPLES/GROUP_BY_DERIVED.PQL 5
options 5
description 5
input section 5
group by 5
output section 5
6. PEQUEL GENERATED PROGRAM 6
7. ABOUT PEQUEL 8
COPYRIGHT 8

SCRIPT NAME

examples/group_by_derived.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 ***2*** 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

```
=> &uc(LOCATION)
```

1.2 *SALES_TOTAL*

Output Field

Description

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

2. CONFIGURATION SETTINGS

2.1 *prefix*
directory pathname prefix.: examples

2.2 *pequeldoc*
generate pod / pdf pequel script Reference Guide.: pdf

2.3 *detail*
Include Pequel Generated Program chapter in Pequeldoc: 1

2.4 *script_name*
script filename: examples/group_by_derived.pql

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.: Group By Derived Example Script

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

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

3. TABLES

4. TABLE INFORMATION SUMMARY

4.1 Table List Sorted By Table Name

5. EXAMPLES/GROUP_BY_DERIVED.PQL

options

```
prefix(examples)
pequeldoc(pdf)
detail(1)
script_name(examples/group_by_derived.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.
```

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 => &uc(LOCATION)
```

group by

```
FIXED_LOC_CODE string
```

output section

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

6. PEQUEL GENERATED PROGRAM

```
#!/usr/bin/perl
#-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
# vim: syntax=perl ts=4 sw=4
#-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
#Generated By: pequel Version 2.4-5, Build: Wednesday November 16 21:56:42 GMT 2005
#            : http://sourceforge.net/projects/pequel/
#Script Name : group_by_derived.pql
#Created On  : Wed Nov 16 14:04:20 2005
#Perl Version: /usr/bin/perl 5.6.1 on solaris
#For         :
#-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
#Options:
#prefix(examples) directory pathname prefix.
#pequeldoc(pdf) generate pod / pdf pequel script Reference Guide.
#detail(1) Include Pequel Generated Program chapter in Pequeldoc
#script_name(examples/group_by_derived.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;
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 _O_FIXED_LOC_CODE    => int    1;
use constant _O_SALES_TOTAL       => int    2;
local $\="\n";
local $,="|";
print STDERR '[examples/group_by_derived.pql ' . localtime() . "] Init";
use constant VERBOSE => int 10000;
use constant LAST_ICELL => int 9;
my @I_VAL;
my %O_VAL;
my $key;
my $_inprecs=0;
&PrintHeader();
print STDERR '[examples/group_by_derived.pql ' . localtime() . "] Start";
use Benchmark;
my $benchmark_start = new Benchmark;
while (<STDIN>)
{
    ++$_inprecs;
    print STDERR '[examples/group_by_derived.pql ' . localtime() . "] $_inprecs records." if ($_inprecs % VERB
OSE == 0);
    chomp;
    @I_VAL = split("[|]", $_);
    $I_VAL[_I_FIXED_LOC_CODE] = uc($I_VAL[_I_LOCATION]);
    $key = ( $I_VAL[_I_FIXED_LOC_CODE] );
    $O_VAL{$key}{_O_FIXED_LOC_CODE} = $I_VAL[_I_FIXED_LOC_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 STDOUT
        $O_VAL{$key}{_O_FIXED_LOC_CODE},
        $O_VAL{$key}{_O_SALES_TOTAL}
    ;
}

close(STDIN);
print STDERR '[examples/group_by_derived.pql ' . localtime() . "] $_inprecs records.";
my $benchmark_end = new Benchmark;
my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
print STDERR '[examples/group_by_derived.pql ' . localtime() . "] Code statistics: @{[timestr($benchmark_timed
iff)]}";
#-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
sub PrintHeader
{
```

```
        local $\="\n";
        local $,="|";
        print STDOUT
            'FIXED_LOC_CODE',
            'SALES_TOTAL'
            ;
}
```

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