



filter_regex.pql

by *Pequel*

sample@youraddress.com

Filer Regex Example Script

2.2

Table of Contents

Filer Regex Example Script

SCRIPT NAME	1
DESCRIPTION	1
1. PROCESS DETAILS	1
1.1 SALES_CODE	1
Description	1
1.2 LOC_DESCRIPT	1
Description	1
Derived Input Field Evaluation	1
1.3 NUM_PRODUCTS	1
Description	1
1.5 PROD_NUM	1
Description	1
Derived Field Evaluation	1
1.6 LOC_NSW	2
Description	2
Derived Field Evaluation	2
1.7 AVG_COST_PRICE_NSW	2
Description	2
Aggregation condition	2
1.8 LOC_WA	2
Description	2
Derived Field Evaluation	2
1.9 AVG_COST_PRICE_WA	2
Description	2
Aggregation condition	2
1.10 LOC_SA	2
Description	2
Derived Field Evaluation	2
1.11 AVG_COST_PRICE_SA	2
Description	2
Aggregation condition	2
2. CONFIGURATION SETTINGS	3
2.1 pequeldoc	3
2.2 detail	3
2.3 script_name	3
2.4 header	3
2.5 optimize	3
2.6 doc_title	3
2.7 doc_email	3
2.8 doc_version	3
3. TABLES	4
3.1 LOC_DESCRIPT	4
Data	4
4. TABLE INFORMATION SUMMARY	5
4.1 Table List Sorted By Table Name	5
5. FILTER_REGEX.PQL	6
options	6
description	6
init table	6
filter	6
sort by	6
group by	6
input section	6

output section	6
6. PEQUEL GENERATED PROGRAM	7
7. ABOUT PEQUEL	10
COPYRIGHT	10

SCRIPT NAME

filter_regex.pql

DESCRIPTION

Demonstrates use of filter and Perl regular expressions. The regular expression can contain Pequel field names macros and table names. This example also demonstrates the use of a simple 'local' table (LOC_DESCRIPT).

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 **11** fields. Fields are delimited by the '|' character.

Input stream is **sorted** by the input field **SALES_CODE** (*string*).

Input records are eliminated (**filtered**) unless **LOCATION** =~ /^NSW\$/^WA\$/^SA\$/.

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

1.1 SALES_CODE

Output Field

Description

Set to input field **SALES_CODE**

1.2 LOC_DESCRIPT

Output Field

Description

Set to input field **LDESCRIPT**

Derived Input Field Evaluation

```
=> %LOC_DESCRIPT(LOCATION)->1 . " in postcode " . %LOC_DESCRIPT(LOCATION)->2
```

1.3 NUM_PRODUCTS

Output Field

Description

Distinct aggregation on input field **PRODUCT_CODE**.

1.5 PROD_NUM

Output Field

Description

Derived (calculated) field.

Derived Field Evaluation

1.6 LOC_NSW

Output Field

Description

Derived (calculated) field.

Derived Field Evaluation**1.7 AVG_COST_PRICE_NSW**

Output Field

Description**Avg** aggregation on input field **COST_PRICE**.**Aggregation condition**

LOCATION eq 'NSW';

1.8 LOC_WA

Output Field

Description

Derived (calculated) field.

Derived Field Evaluation**1.9 AVG_COST_PRICE_WA**

Output Field

Description**Avg** aggregation on input field **COST_PRICE**.**Aggregation condition**

LOCATION eq 'WA';

1.10 LOC_SA

Output Field

Description

Derived (calculated) field.

Derived Field Evaluation**1.11 AVG_COST_PRICE_SA**

Output Field

Description**Avg** aggregation on input field **COST_PRICE**.**Aggregation condition**

LOCATION eq 'SA';

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: filter_regex.pql

2.4 *header*

write header record to output.: 1

2.5 *optimize*

optimize generated code.: 1

2.6 *doc_title*

document title.: Filer Regex Example Script

2.7 *doc_email*

document email entry.: sample@youraddress.com

2.8 *doc_version*

document version for pequel script.: 2.2

3. TABLES

3.1 *LOC_DESCRIPT*

Table Type: *local*

Data

MEL — Melbourne 2022 07

NSW — New South Wales 2061 02

WA — Western Australia 5008 07

SYD — Sydney 2000

SA — South Australia 8078

NT — Northern Territory 6509

QLD — Queensland 3021

VIC — Victoria 2343

PER — Perth 7652

ALIC — Alice Springs 8978

4. TABLE INFORMATION SUMMARY

4.1 Table List Sorted By Table Name

LOC_DESCRIPT — 1 (*local*)

5. FILTER_REGEX.PQL

options

```
pequeldoc(pdf)
detail(1)
script_name(filter_regex.pql)
header(1)
optimize(1)
doc_title(Filer Regex Example Script)
doc_email(sample@youraddress.com)
doc_version(2.2)
```

description

Demonstrates use of filter and Perl regular expressions. The regular expression can contain Pequel field names macros and table names. This example also demonstrates the use of a simple 'local' table (LOC_DESCRIPT).

init table

```
LOC_DESCRIPT MEL Melbourne 2022 07
LOC_DESCRIPT NSW New South Wales 2061 02
LOC_DESCRIPT WA Western Australia 5008 07
LOC_DESCRIPT SYD Sydney 2000
LOC_DESCRIPT SA South Australia 8078
LOC_DESCRIPT NT Northern Territory 6509
LOC_DESCRIPT QLD Queensland 3021
LOC_DESCRIPT VIC Victoria 2343
LOC_DESCRIPT PER Perth 7652
LOC_DESCRIPT ALIC Alice Springs 8978
```

filter

```
LOCATION =~ /^NSW$|^WA$|^SA$/
```

sort by

```
SALES_CODE string
```

group by

```
SALES_CODE string
```

input section

```
PRODUCT_CODE
COST_PRICE
DESCRIPTION
SALES_CODE
SALES_PRICE
SALES_QTY
SALES_DATE
LOCATION
LDESCRIPT => %LOC_DESCRIPT(LOCATION)->1 . " in postcode " . %LOC_DESCRIPT(LOCATION) \
->2
```

output section

string	SALES_CODE	SALES_CODE
string	LOC_DESCRIPT	LDESCRIPT
numeric	NUM_PRODUCTS	distinct PRODUCT_CODE
string	_PRODUCT_CODE	PRODUCT_CODE
string	PROD_NUM	= _PRODUCT_CODE . "-" . NUM_PRODUCTS
string	LOC_NSW	= %LOC_DESCRIPT(NSW)->1
numeric	AVG_COST_PRICE_NSW	avg COST_PRICE where LOCATION eq 'NSW'
string	LOC_WA	= %LOC_DESCRIPT(WA)->1
numeric	AVG_COST_PRICE_WA	avg COST_PRICE where LOCATION eq 'WA'
string	LOC_SA	= %LOC_DESCRIPT(SA)->1
numeric	AVG_COST_PRICE_SA	avg COST_PRICE where LOCATION eq 'SA'

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 : filter_regex.pql
#Created On  : Tue Sep 13 10:20:58 2005
#For         :
#-----
#Options:
#pequeldoc(pdf) generate pod / pdf pequel script Reference Guide.
#detail(1) Include Pequel Generated Program chapter in Pequeldoc
#script_name(filter_regex.pql) script filename
#header(1) write header record to output.
#optimize(1) optimize generated code.
#doc_title(Filer Regex 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 "[filter_regex.pql ' . localtime() . "] Init";
use constant VERBOSE => int 10000;
use constant LAST_ICELL => int 8;
my @I_VAL;
my @O_VAL;
my %DISTINCT;
my %AVERAGE;
my $key__I_SALES_CODE;
my $previous_key__I_SALES_CODE = undef;
foreach my $f (1..11) { $O_VAL[$f] = undef; }
my $_TABLE_LOC_DESCRIPT = &InitLookupLOC_DESCRIPT; # ref to %$_LOC_DESCRIPT 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_LDESCRIPT         => int 8;
use constant _O_SALES_CODE        => int 1;
use constant _O_LOC_DESCRIPT      => int 2;
use constant _O_NUM_PRODUCTS      => int 3;
use constant _O_PRODUCT_CODE      => int 4;
use constant _O_PROD_NUM          => int 5;
use constant _O_LOC_NSW           => int 6;
use constant _O_AVG_COST_PRICE_NSW => int 7;
use constant _O_LOC_WA            => int 8;
use constant _O_AVG_COST_PRICE_WA => int 9;
use constant _O_LOC_SA            => int 10;
use constant _O_AVG_COST_PRICE_SA => int 11;
use constant _T_LOC_DESCRIPT_FLD_1 => int 0;
use constant _T_LOC_DESCRIPT_FLD_2 => int 1;
use constant _T_LOC_DESCRIPT_FLD_3 => int 2;
use constant _I_LOC_DESCRIPT_LOCATION_FLD_KEY => int 9;
use constant _I_LOC_DESCRIPT_LOCATION_FLD_1 => int 10;
use constant _I_LOC_DESCRIPT_LOCATION_FLD_2 => int 11;
use constant _I_LOC_DESCRIPT_LOCATION_FLD_3 => int 12;
open(DATA, q{cat - | sort -t'|' -y -k 4,4 |}) || die "Cannot open input: $!";
&PrintHeader();
print STDERR "[filter_regex.pql ' . localtime() . "] Start";
use Benchmark;
my $benchmark_start = new Benchmark;
while (<DATA>)
{
    print STDERR "[filter_regex.pql ' . localtime() . "] $. records." if ($. % VERBOSE == 0);
    chomp;
    @I_VAL = split("[|]", $_);
    next unless ($I_VAL[_I_LOCATION] =~ /^NSW$|^WA$|^SA$/);
    $key__I_SALES_CODE = $I_VAL[_I_SALES_CODE];
    if (!defined($previous_key__I_SALES_CODE))
    {
        $previous_key__I_SALES_CODE = $key__I_SALES_CODE;
    }

    elsif ($previous_key__I_SALES_CODE ne $key__I_SALES_CODE)
    {
        $O_VAL[_O_PROD_NUM] = $O_VAL[_O_PRODUCT_CODE] . "-" . $O_VAL[_O_NUM_PRODUCTS];
        $O_VAL[_O_LOC_NSW] = ${$_TABLE_LOC_DESCRIPT{qq{NSW}}}{_T_LOC_DESCRIPT_FLD_1};
        $O_VAL[_O_AVG_COST_PRICE_NSW] = ($AVERAGE{$_O_AVG_COST_PRICE_NSW}{_COUNT} == 0 ? 0 : $AVERAGE{$_O_AVG_COST_PRICE_NSW});
    }
}
```

```

ST_PRICE_NSW}{_SUM} / $AVERAGE{_O_AVG_COST_PRICE_NSW}{_COUNT});
    $O_VAL[_O_LOC_WA] = ${$_TABLE_LOC_DESCRIPTOR{qq{WA}}}{_T_LOC_DESCRIPTOR_FLD_1};
    $O_VAL[_O_AVG_COST_PRICE_WA] = ($AVERAGE{_O_AVG_COST_PRICE_WA}{_COUNT} == 0 ? 0 : $AVERAGE{_O_AVG_COST_PRICE_WA}{_SUM} / $AVERAGE{_O_AVG_COST_PRICE_WA}{_COUNT});
    $O_VAL[_O_LOC_SA] = ${$_TABLE_LOC_DESCRIPTOR{qq{SA}}}{_T_LOC_DESCRIPTOR_FLD_1};
    $O_VAL[_O_AVG_COST_PRICE_SA] = ($AVERAGE{_O_AVG_COST_PRICE_SA}{_COUNT} == 0 ? 0 : $AVERAGE{_O_AVG_COST_PRICE_SA}{_SUM} / $AVERAGE{_O_AVG_COST_PRICE_SA}{_COUNT});
    print
        $O_VAL[_O_SALES_CODE],
        $O_VAL[_O_LOC_DESCRIPTOR],
        $O_VAL[_O_NUM_PRODUCTS],
        $O_VAL[_O_PROD_NUM],
        $O_VAL[_O_LOC_NSW],
        $O_VAL[_O_AVG_COST_PRICE_NSW],
        $O_VAL[_O_LOC_WA],
        $O_VAL[_O_AVG_COST_PRICE_WA],
        $O_VAL[_O_LOC_SA],
        $O_VAL[_O_AVG_COST_PRICE_SA]
    ;
    $previous_key__I_SALES_CODE = $key__I_SALES_CODE;
    @O_VAL = undef;
    %DISTINCT = undef;
    %AVERAGE = undef;
}

    $O_VAL[_O_SALES_CODE] = $I_VAL[_I_SALES_CODE];
    $I_VAL[_I_LDESCRIPTOR] = ${$_TABLE_LOC_DESCRIPTOR{qq{$I_VAL[_I_LOCATION]}}}{_T_LOC_DESCRIPTOR_FLD_1} . " in pos
tcode " . ${$_TABLE_LOC_DESCRIPTOR{qq{$I_VAL[_I_LOCATION]}}}{_T_LOC_DESCRIPTOR_FLD_2};
    $O_VAL[_O_LOC_DESCRIPTOR] = $I_VAL[_I_LDESCRIPTOR];
    $O_VAL[_O_NUM_PRODUCTS]++ if (defined($I_VAL[_I_PRODUCT_CODE]) && ++$DISTINCT{_O_NUM_PRODUCTS}{qq{$I_VAL[_I_PRODUCT_CODE]}} == 1);
    $O_VAL[_O_PRODUCT_CODE] = $I_VAL[_I_PRODUCT_CODE];

    if ($I_VAL[_I_LOCATION] eq 'NSW') {
        $AVERAGE{_O_AVG_COST_PRICE_NSW}{_SUM} += $I_VAL[_I_COST_PRICE];
        $AVERAGE{_O_AVG_COST_PRICE_NSW}{_COUNT}++;
    }
    elsif ($I_VAL[_I_LOCATION] eq 'SA') {
        $AVERAGE{_O_AVG_COST_PRICE_SA}{_SUM} += $I_VAL[_I_COST_PRICE];
        $AVERAGE{_O_AVG_COST_PRICE_SA}{_COUNT}++;
    }
    elsif ($I_VAL[_I_LOCATION] eq 'WA') {
        $AVERAGE{_O_AVG_COST_PRICE_WA}{_SUM} += $I_VAL[_I_COST_PRICE];
        $AVERAGE{_O_AVG_COST_PRICE_WA}{_COUNT}++;
    }
}

$O_VAL[_O_PROD_NUM] = $O_VAL[_O_PRODUCT_CODE] . "-" . $O_VAL[_O_NUM_PRODUCTS];
$O_VAL[_O_LOC_NSW] = ${$_TABLE_LOC_DESCRIPTOR{qq{NSW}}}{_T_LOC_DESCRIPTOR_FLD_1};
$O_VAL[_O_AVG_COST_PRICE_NSW] = ($AVERAGE{_O_AVG_COST_PRICE_NSW}{_COUNT} == 0 ? 0 : $AVERAGE{_O_AVG_COST_PRICE_NSW}{_SUM} / $AVERAGE{_O_AVG_COST_PRICE_NSW}{_COUNT});
$O_VAL[_O_LOC_WA] = ${$_TABLE_LOC_DESCRIPTOR{qq{WA}}}{_T_LOC_DESCRIPTOR_FLD_1};
$O_VAL[_O_AVG_COST_PRICE_WA] = ($AVERAGE{_O_AVG_COST_PRICE_WA}{_COUNT} == 0 ? 0 : $AVERAGE{_O_AVG_COST_PRICE_WA}{_SUM} / $AVERAGE{_O_AVG_COST_PRICE_WA}{_COUNT});
$O_VAL[_O_LOC_SA] = ${$_TABLE_LOC_DESCRIPTOR{qq{SA}}}{_T_LOC_DESCRIPTOR_FLD_1};
$O_VAL[_O_AVG_COST_PRICE_SA] = ($AVERAGE{_O_AVG_COST_PRICE_SA}{_COUNT} == 0 ? 0 : $AVERAGE{_O_AVG_COST_PRICE_SA}{_SUM} / $AVERAGE{_O_AVG_COST_PRICE_SA}{_COUNT});
print
    $O_VAL[_O_SALES_CODE],
    $O_VAL[_O_LOC_DESCRIPTOR],
    $O_VAL[_O_NUM_PRODUCTS],
    $O_VAL[_O_PROD_NUM],
    $O_VAL[_O_LOC_NSW],
    $O_VAL[_O_AVG_COST_PRICE_NSW],
    $O_VAL[_O_LOC_WA],
    $O_VAL[_O_AVG_COST_PRICE_WA],
    $O_VAL[_O_LOC_SA],
    $O_VAL[_O_AVG_COST_PRICE_SA]
;
print STDERR "[filter_regex.pql ' . localtime() . "] $. records.";
my $benchmark_end = new Benchmark;
my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
print STDERR "[filter_regex.pql ' . localtime() . "] Code statistics: @([timestr($benchmark_timediff)])";
#-----
##### Table LOC_DESCRIPTOR --> Type :Pequel::Type::Table::Local #####
sub InitLookupLOC_DESCRIPTOR
{
    my $_TABLE_LOC_DESCRIPTOR;
    $_TABLE_LOC_DESCRIPTOR =
    (
        'ALIC' => ['Alice Springs', '8978', ''],
        'MEL' => ['Melbourne', '2022', '07'],
        'NSW' => ['New South Wales', '2061', '02'],
        'NT' => ['Northern Territory', '6509', ''],
        'PER' => ['Perth', '7652', ''],
    )
}

```

```
        'QLD' => ['Queensland', '3021', ''],
        'SA'  => ['South Australia', '8078', ''],
        'SYD' => ['Sydney', '2000', ''],
        'VIC' => ['Victoria', '2343', ''],
        'WA'  => ['Western Australia', '5008', '07']
    );
    return \%_TABLE_LOC_DESCRIPT;
}

sub PrintHeader
{
    local $\="\n";
    local $,="|";
    print
        'SALES_CODE',
        'LOC_DESCRIPT',
        'NUM_PRODUCTS',
        'PROD_NUM',
        'LOC_NSW',
        'AVG_COST_PRICE_NSW',
        'LOC_WA',
        'AVG_COST_PRICE_WA',
        'LOC_SA',
        'AVG_COST_PRICE_SA'
    ;
}
```

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

