



examples/transfer_option.pql

by *Pequel*

sample@youraddress.com

Transfer Option Example Script

2.2

Table of Contents

Transfer Option Example Script

| | |
|-------------------------------------|----|
| SCRIPT NAME | 1 |
| DESCRIPTION | 1 |
| 1. PROCESS DETAILS | 1 |
| 1.1 PRODUCT_CODE | 1 |
| 1.2 COST_PRICE | 1 |
| 1.3 DESCRIPTION | 1 |
| 1.4 SALES_CODE | 1 |
| 1.5 SALES_PRICE | 1 |
| 1.6 SALES_QTY | 1 |
| 1.7 SALES_DATE | 1 |
| 1.8 LOCATION | 1 |
| 1.9 LDESCRIPT | 1 |
| Derived Field Evaluation | 1 |
| 1.10 ZIPCODE | 1 |
| Derived Field Evaluation | 1 |
| 1.11 SALES_TOTAL | 2 |
| Derived Field Evaluation | 2 |
| 1.12 SALES_Q1 | 2 |
| Derived Field Evaluation | 2 |
| 1.13 SALES_Q2 | 2 |
| Derived Field Evaluation | 2 |
| 1.14 SALES_Q3 | 2 |
| Derived Field Evaluation | 2 |
| 1.15 SALES_Q4 | 2 |
| Derived Field Evaluation | 2 |
| 2. CONFIGURATION SETTINGS | 3 |
| 2.1 prefix | 3 |
| 2.2 pequeldoc | 3 |
| 2.3 detail | 3 |
| 2.4 script_name | 3 |
| 2.5 header | 3 |
| 2.6 optimize | 3 |
| 2.7 transfer | 3 |
| 2.8 discard_header | 3 |
| 2.9 default_datatype | 3 |
| 2.10 doc_title | 3 |
| 2.11 doc_email | 3 |
| 2.12 doc_version | 3 |
| 3. TABLES | 4 |
| 3.1 LOC_DESCRIPT | 4 |
| Data | 4 |
| 3.2 LOC_DESCRIPTION | 4 |
| Data | 4 |
| 4. TABLE INFORMATION SUMMARY | 5 |
| 4.1 Table List Sorted By Table Name | 5 |
| 5. EXAMPLES/TRANSFER_OPTION.PQL | 6 |
| options | 6 |
| description | 6 |
| init table | 6 |
| input section | 6 |
| 6. PEQUEL GENERATED PROGRAM | 7 |
| 7. ABOUT PEQUEL | 10 |
| COPYRIGHT | 10 |

SCRIPT NAME

examples/transfer_option.pql

DESCRIPTION

Demonstrates use of 'transfer' option and &date() macro. Note the &date() macro converts date values to YYYYMMDD format.

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

1.1 PRODUCT_CODE

Input Field

1.2 COST_PRICE

Input Field

1.3 DESCRIPTION

Input Field

1.4 SALES_CODE

Input Field

1.5 SALES_PRICE

Input Field

1.6 SALES_QTY

Input Field

1.7 SALES_DATE

Input Field

1.8 LOCATION

Input Field

1.9 LDESCRIPT

Input Derived Field

Derived Field Evaluation

```
=> %LOC_DESCRIPT(LOCATION)
```

1.10 ZIPCODE

Input Derived Field

Derived Field Evaluation

```
=> %LOC_DESCRIPTION(LOCATION)->2
```

1.11 SALES_TOTAL

Input Derived Field

Derived Field Evaluation

```
=> SALES_PRICE * SALES_QTY
```

1.12 SALES_Q1

Input Derived Field

Derived Field Evaluation

```
=> (&date(SALES_DATE,YYMMDD) >= 20020101 && &date(SALES_DATE) <= 20020331)
?   SALES_TOTAL
:   0.0
```

1.13 SALES_Q2

Input Derived Field

Derived Field Evaluation

```
=> (&date(SALES_DATE) >= 20020401 && &date(SALES_DATE) <= 30030631)
?   SALES_TOTAL
:   0.0
```

1.14 SALES_Q3

Input Derived Field

Derived Field Evaluation

```
=> (&date(SALES_DATE) >= 20020701 && &date(SALES_DATE) <= 20020931)
?   SALES_TOTAL
:   0.0
```

1.15 SALES_Q4

Input Derived Field

Derived Field Evaluation

```
=> (&date(SALES_DATE) >= 20021001 && &date(SALES_DATE) <= 20021231)
?   SALES_TOTAL
:   0.0
```

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/transfer_option.pql

2.5 *header*

write header record to output.: 1

2.6 *optimize*

optimize generated code.: 1

2.7 *transfer*

Copy input (including calculated fields) to output.: 1

2.8 *discard_header*

Input file has header record - must be discarded.: 1

2.9 *default_datetype*

default date format: DD/MM/YYYY

2.10 *doc_title*

document title.: Transfer Option Example Script

2.11 *doc_email*

document email entry.: sample@youraddress.com

2.12 *doc_version*

document version for pequel script.: 2.2

3. TABLES

3.1 *LOC_DESCRIPTOR*

Table Type: *local*

Data

NSW — New South Wales

WA — Western Australia

SYD — Sydney

MEL — Melbourne

SA — South Australia

NT — Northern Territory

QLD — Queensland

VIC — Victoria

PER — Perth

ALIC — Alice Springs

3.2 *LOC_DESCRIPTION*

Table Type: *local*

Data

WA — Western Australia 4001

NSW — New South Wales 2061

SA — South Australia 3023

QLD — Queensland 6099

4. TABLE INFORMATION SUMMARY

4.1 Table List Sorted By Table Name

LOC_DESCRIPT — **1** (*local*)

LOC_DESCRIPTION — **2** (*local*)

5. EXAMPLES/TRANSFER_OPTION.PQL

options

```

prefix(examples)
pequeldoc(pdf)
detail(1)
script_name(examples/transfer_option.pql)
header(1)
optimize(1)
transfer(1)
discard_header(1)
default_datatype(DD/MM/YYYY)
doc_title(Transfer Option Example Script)
doc_email(sample@youraddress.com)
doc_version(2.2)

```

description

Demonstrates use of 'transfer' option
and &date() macro.
Note the &date() macro converts date values to YYYYMMDD format.

init table

```

LOC_DESCRIPTOR NSW New South Wales
LOC_DESCRIPTOR WA Western Australia
LOC_DESCRIPTOR SYD Sydney
LOC_DESCRIPTOR MEL Melbourne
LOC_DESCRIPTOR SA South Australia
LOC_DESCRIPTOR NT Northern Territory
LOC_DESCRIPTOR QLD Queensland
LOC_DESCRIPTOR VIC Victoria
LOC_DESCRIPTOR PER Perth
LOC_DESCRIPTOR ALIC Alice Springs

LOC_DESCRIPTION WA Western Australia 4001
LOC_DESCRIPTION NSW New South Wales 2061
LOC_DESCRIPTION SA South Australia 3023
LOC_DESCRIPTION QLD Queensland 6099

```

input section

```

PRODUCT_CODE
COST_PRICE
DESCRIPTION
SALES_CODE
SALES_PRICE
SALES_QTY
SALES_DATE
LOCATION
LDESCRIPTION => %LOC_DESCRIPTOR(LOCATION)

ZIPCODE => %LOC_DESCRIPTION(LOCATION)->2

SALES_TOTAL => SALES_PRICE * SALES_QTY

SALES_Q1 => (&date(SALES_DATE,YYYYMMDD) >= 20020101 && &date(SALES_DATE) <= 20020331)
? SALES_TOTAL
: 0.0

SALES_Q2 => (&date(SALES_DATE) >= 20020401 && &date(SALES_DATE) <= 30030631)
? SALES_TOTAL
: 0.0

SALES_Q3 => (&date(SALES_DATE) >= 20020701 && &date(SALES_DATE) <= 20020931)
? SALES_TOTAL
: 0.0

SALES_Q4 => (&date(SALES_DATE) >= 20021001 && &date(SALES_DATE) <= 20021231)
? SALES_TOTAL
: 0.0

```

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 : transfer_option.pql
#Created On  : Wed Nov 16 14:22:15 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/transfer_option.pql) script filename
#header(1) write header record to output.
#optimize(1) optimize generated code.
#transfer(1) Copy input (including calculated fields) to output.
#discard_header(1) Input file has header record - must be discarded.
#default_datetype(DD/MM/YYYY) default date format
#doc_title(Transfer Option 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_LDESCRIPT      => int    8;
use constant _I_ZIPCODE        => int    9;
use constant _I_SALES_TOTAL    => int   10;
use constant _I_SALES_Q1      => int   11;
use constant _I_SALES_Q2      => int   12;
use constant _I_SALES_Q3      => int   13;
use constant _I_SALES_Q4      => int   14;
use constant _T_LOC_DESCRIPT_FLD_1  => int    0;
use constant _T_LOC_DESCRIPTION_FLD_1  => int    0;
use constant _T_LOC_DESCRIPTION_FLD_2  => int    1;
use constant _I_LOC_DESCRIPT_LOCATION_FLD_KEY  => int   15;
use constant _I_LOC_DESCRIPT_LOCATION_FLD_1  => int   16;
use constant _I_LOC_DESCRIPT_LOCATION_FLD_KEY => int   17;
use constant _I_LOC_DESCRIPTION_LOCATION_FLD_1  => int   18;
use constant _I_LOC_DESCRIPTION_LOCATION_FLD_2  => int   19;
local $\\="\\n";
local $,="|";
print STDERR '[examples/transfer_option.pql ' . localtime() . "] Init";
use constant VERBOSE => int 10000;
use constant LAST_ICELL => int 14;
my @I_VAL;
my @O_VAL;
my $_inprecs=0;
my %MONTH_NUM =
(
    JAN => '01',
    FEB => '02',
    MAR => '03',
    APR => '04',
    MAY => '05',
    JUN => '06',
    JUL => '07',
    AUG => '08',
    SEP => '09',
    OCT => '10',
    NOV => '11',
    DEC => '12'
);
my %MONTH_NAME; foreach my $m (keys %MONTH_NUM) { $MONTH_NAME{$MONTH_NUM{$m}} = $m; }
my $_TABLE_LOC_DESCRIPT = &InitLookupLOC_DESCRIPT; # ref to %$LOC_DESCRIPT hash
my $_TABLE_LOC_DESCRIPTION = &InitLookupLOC_DESCRIPTION; # ref to %$LOC_DESCRIPTION hash
&PrintHeader();
print STDERR '[examples/transfer_option.pql ' . localtime() . "] Start";
use Benchmark;
my $benchmark_start = new Benchmark;
my $discard_header = <STDIN>;
```

```

while (<STDIN>)
{
    ++$_inprec;
    print STDERR '[examples/transfer_option.pql ' . localtime() . "] $_inprec records." if ($_inprec % VERBO
SE == 0);
    chomp;
    @I_VAL = split("[|]", $_);
    $I_VAL[_I_LDESCRPT] = $$TABLE_LOC_DESCRIPT{qq{$I_VAL[_I_LOCATION]}};
    $I_VAL[_I_ZIPCODE] = ${$TABLE_LOC_DESCRIPTION{qq{$I_VAL[_I_LOCATION]}}}{_T_LOC_DESCRIPTION_FLD_2};
    $I_VAL[_I_SALES_TOTAL] = $I_VAL[_I_SALES_PRICE] * $I_VAL[_I_SALES_QTY];
    $I_VAL[_I_SALES_Q1] = (scalar((int(substr(qq{$I_VAL[_I_SALES_DATE]}, 0, 2)) < 20 ? '20' : '19') . substr(q
q{$I_VAL[_I_SALES_DATE]}, 0, 2). $MONTH_NUM(substr(qq{$I_VAL[_I_SALES_DATE]}, 2, 3). substr(qq{$I_VAL[_I_SALE
S_DATE]}, 5, 2)) >= 20020101 && scalar(substr(qq{$I_VAL[_I_SALES_DATE]}, 6, 4). substr(qq{$I_VAL[_I_SALES_DATE
]}, 3, 2). substr(qq{$I_VAL[_I_SALES_DATE]}, 0, 2)) <= 20020331) ? $I_VAL[_I_SALES_TOTAL] : 0.0;
    $I_VAL[_I_SALES_Q2] = (scalar(substr(qq{$I_VAL[_I_SALES_DATE]}, 6, 4). substr(qq{$I_VAL[_I_SALES_DATE]}, 3
, 2). substr(qq{$I_VAL[_I_SALES_DATE]}, 0, 2)) >= 20020401 && scalar(substr(qq{$I_VAL[_I_SALES_DATE]}, 6, 4).
substr(qq{$I_VAL[_I_SALES_DATE]}, 3, 2). substr(qq{$I_VAL[_I_SALES_DATE]}, 0, 2)) <= 30030631) ? $I_VAL[_I_SAL
ES_TOTAL] : 0.0;
    $I_VAL[_I_SALES_Q3] = (scalar(substr(qq{$I_VAL[_I_SALES_DATE]}, 6, 4). substr(qq{$I_VAL[_I_SALES_DATE]}, 3
, 2). substr(qq{$I_VAL[_I_SALES_DATE]}, 0, 2)) >= 20020701 && scalar(substr(qq{$I_VAL[_I_SALES_DATE]}, 6, 4).
substr(qq{$I_VAL[_I_SALES_DATE]}, 3, 2). substr(qq{$I_VAL[_I_SALES_DATE]}, 0, 2)) <= 20020931) ? $I_VAL[_I_SAL
ES_TOTAL] : 0.0;
    $I_VAL[_I_SALES_Q4] = (scalar(substr(qq{$I_VAL[_I_SALES_DATE]}, 6, 4). substr(qq{$I_VAL[_I_SALES_DATE]}, 3
, 2). substr(qq{$I_VAL[_I_SALES_DATE]}, 0, 2)) >= 20021001 && scalar(substr(qq{$I_VAL[_I_SALES_DATE]}, 6, 4).
substr(qq{$I_VAL[_I_SALES_DATE]}, 3, 2). substr(qq{$I_VAL[_I_SALES_DATE]}, 0, 2)) <= 20021231) ? $I_VAL[_I_SAL
ES_TOTAL] : 0.0;
    print STDOUT
        @I_VAL[0..LAST_ICELL]
    ;
}

close(STDIN);
print STDERR '[examples/transfer_option.pql ' . localtime() . "] $_inprec records.";
my $benchmark_end = new Benchmark;
my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
print STDERR '[examples/transfer_option.pql ' . localtime() . "] Code statistics: @{{timestr($benchmark_timedi
ff)}}";
#-+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
#+++++ Table LOC_DESCRIPTOR --> Type :ETL::Pequel::Type::Table::Local +++++
sub InitLookupLOC_DESCRIPTOR
{
    my %_TABLE_LOC_DESCRIPTOR;
    %_TABLE_LOC_DESCRIPTOR =
    (
        'ALIC' => 'Alice Springs',
        'MEL' => 'Melbourne',
        'NSW' => 'New South Wales',
        'NT' => 'Northern Territory',
        'PER' => 'Perth',
        'QLD' => 'Queensland',
        'SA' => 'South Australia',
        'SYD' => 'Sydney',
        'VIC' => 'Victoria',
        'WA' => 'Western Australia'
    );
    return \%_TABLE_LOC_DESCRIPTOR;
}

#+++++ Table LOC_DESCRIPTION --> Type :ETL::Pequel::Type::Table::Local +++++
sub InitLookupLOC_DESCRIPTION
{
    my %_TABLE_LOC_DESCRIPTION;
    %_TABLE_LOC_DESCRIPTION =
    (
        'NSW' => ['New South Wales', '2061'],
        'QLD' => ['Queensland', '6099'],
        'SA' => ['South Australia', '3023'],
        'WA' => ['Western Australia', '4001']
    );
    return \%_TABLE_LOC_DESCRIPTION;
}

sub PrintHeader
{
    local $\="\n";
    local $,="|";
    print STDOUT
        'PRODUCT_CODE',
        'COST_PRICE',
        'DESCRIPTION',
        'SALES_CODE',
        'SALES_PRICE',
        'SALES_QTY',
        'SALES_DATE',
        'LOCATION',

```

```
        'LDESCRIPT' ,  
        'ZIPCODE' ,  
        'SALES_TOTAL' ,  
        'SALES_Q1' ,  
        'SALES_Q2' ,  
        'SALES_Q3' ,  
        'SALES_Q4'  
    ;  
}
```

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

