



# examples/copy\_record.pql

by *Pequel*

---

[sample@youraddress.com](mailto:sample@youraddress.com)

## Copy Record Example Script

2.3



# Table of Contents

## Copy Record Example Script

SCRIPT NAME	1
DESCRIPTION	1
1. PROCESS DETAILS	1
1.1 LOCATION	1
Description	1
Derived Input Field Evaluation	1
1.2 DESCRIPTION	1
Description	1
Derived Input Field Evaluation	1
1.3 SALES_TOTAL	1
Description	1
2. CONFIGURATION SETTINGS	2
2.1 pequeldoc	2
2.2 detail	2
2.3 prefix	2
2.4 script_name	2
2.5 input_file	2
2.6 optimize	2
2.7 doc_title	2
2.8 doc_email	2
2.9 doc_version	2
3. TABLES	3
3.1 LOC_DESCRIPT	3
Data	3
4. TABLE INFORMATION SUMMARY	4
4.1 Table List Sorted By Table Name	4
5. EXAMPLES/COPY_RECORD.PQL	5
options	5
init table	5
input section	5
copy record(pequel:copy_record_WA.pql)	5
copy record(pequel:copy_record_SA.pql)	5
copy record(pequel:copy_record_NSW.pql)	5
copy record(pequel:copy_record_VIC.pql)	5
copy record(pequel:copy_record_NT.pql)	5
filter	5
sort by	5
group by	5
output section	5
sort output	5
6. PEQUEL GENERATED PROGRAM	6
7. ABOUT PEQUEL	15
COPYRIGHT	15



## SCRIPT NAME

examples/copy\_record.pql

## DESCRIPTION

### 1. PROCESS DETAILS

Input records are read from chain\_pequel\_pt1.pql. The input record contains **3** 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 stream is **sorted** by the input field **LOCATION** (*string*).

Input records are eliminated (**filtered**) unless **LOCATION eq 'WA' || LOCATION eq 'SA' || LOCATION eq 'NSW' || LOCATION eq 'VIC' || LOCATION eq 'NT'**.

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

#### 1.1 LOCATION

Output Field

##### **Description**

Set to input field **LOCATION\_DESC**

##### **Derived Input Field Evaluation**

=> %LOC\_DESCRIPTOR(LOCATION)

#### 1.2 DESCRIPTION

Output Field

##### **Description**

Set to input field **DESCRIPTION**

##### **Derived Input Field Evaluation**

=> 'State Total'

#### 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 *prefix*

directory pathname prefix.: examples

### 2.4 *script\_name*

script filename: examples/copy\_record.pql

### 2.5 *input\_file*

input data filename: chain\_pequel\_pt1.pql

### 2.6 *optimize*

optimize generated code.: 1

### 2.7 *doc\_title*

document title.: Copy Record Example Script

### 2.8 *doc\_email*

document email entry.: sample@youraddress.com

### 2.9 *doc\_version*

document version for pequel script.: 2.3

### 3. TABLES

#### 3.1 *LOC\_DESCRIPT*

Table Type: *local*

***Data***

NSW — New South Wales

WA — Western Australia

SA — South Australia

NT — Northern Territory

QLD — Queensland

VIC — Victoria

## 4. TABLE INFORMATION SUMMARY

### 4.1 Table List Sorted By Table Name

LOC\_DESCRIPT — 1 (*local*)



## 5. EXAMPLES/COPY\_RECORD.PQL

### options

```
pequeldoc(pdf)
detail(1)
prefix(examples)
script_name(examples/copy_record.pql)
input_file(chain_pequel_pt1.pql)
optimize(1)
doc_title(Copy Record Example Script)
doc_email(sample@youraddress.com)
doc_version(2.3)
```

### init table

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

### input section

```
LOCATION
PRODUCT_CODE
SALES_TOTAL
LOCATION_DESC => %LOC_DESCRIPTOR(LOCATION)

DESCRIPTION => 'State Total'
```

### copy record(pequel:copy\_record\_WA.pql)

```
LOCATION eq 'WA'
```

### copy record(pequel:copy\_record\_SA.pql)

```
LOCATION eq 'SA'
```

### copy record(pequel:copy\_record\_NSW.pql)

```
LOCATION eq 'NSW'
```

### copy record(pequel:copy\_record\_VIC.pql)

```
LOCATION eq 'VIC'
```

### copy record(pequel:copy\_record\_NT.pql)

```
LOCATION eq 'NT'
```

### filter

```
LOCATION eq 'WA' || LOCATION eq 'SA' || LOCATION eq 'NSW' || LOCATION eq 'VIC' || LOCATION eq 'NT'
```

### sort by

```
LOCATION string
```

### group by

```
LOCATION string
```

### output section

```
string    LOCATION    LOCATION_DESC
string    DESCRIPTION DESCRIPTION
decimal   SALES_TOTAL sum SALES_TOTAL
```

### sort output

```
LOCATION string
SALES_TOTAL numeric des
```

## 6. PEQUEL GENERATED PROGRAM

```
# vim: syntax=perl ts=4 sw=4
#-----
#Generated By: pequel Version 2.3-4, Build: Wednesday October 12 23:16:49 BST 2005
#           : https://sourceforge.net/projects/pequel/
#Script Name : examples/copy_record.pql
#Created On  : Wed Oct 12 15:27:36 2005
#For         :
#-----
#Options:
#pequeldoc(pdf) generate pod / pdf pequel script Reference Guide.
#detail(1) Include Pequel Generated Program chapter in Pequeldoc
#prefix(examples) directory pathname prefix.
#script_name(examples/copy_record.pql) script filename
#input_file(chain_pequel_pt1.pql) input data filename
#optimize(1) optimize generated code.
#doc_title(Copy Record Example Script) document title.
#doc_email(sample@youraddress.com) document email entry.
#doc_version(2.3) document version for pequel script.
#-----
use strict;
use Fcntl ':flock';
use constant _I_LOCATION      => int    0;
use constant _I_PRODUCT_CODE  => int    1;
use constant _I_SALES_TOTAL   => int    2;
use constant _I_LOCATION_DESC => int    3;
use constant _I_DESCRIPTION   => int    4;
use constant _O_LOCATION      => int    1;
use constant _O_DESCRIPTION   => int    2;
use constant _O_SALES_TOTAL   => int    3;
use constant _T_LOC_DESCRIPTOR_FLD_1 => int    0;
use constant _I_LOC_DESCRIPTOR_LOCATION_FLD_KEY => int    5;
use constant _I_LOC_DESCRIPTOR_LOCATION_FLD_1 => int    6;
local $\\="\\n"; local $,="|";
print STDERR "[examples/copy_record.pql ' . localtime() . "] Init";
use constant VERBOSE => int 10000;
use constant LAST_ICELL => int 4;
my @I_VAL;
my @O_VAL;
my $key__I_LOCATION;
my $previous_key__I_LOCATION = undef;
foreach my $f (1..3) { $O_VAL[$f] = undef; }
my $TABLE_LOC_DESCRIPTOR = &InitLookupLOC_DESCRIPTOR; # ref to %$LOC_DESCRIPTOR hash
if (open(CHAIN_PEQUEL_PT1, '-|') == 0) # Fork -- read from child
{
    &p_execPequelCHAIN_PEQUEL_PT1::execPequelCHAIN_PEQUEL_PT1;
    exit(0);
}

open(STDOUT, '|-', q{sort -t'|' -y -k 1,1 -k 3nr,3nr 2>/dev/null |});
if (open(COPY_COPY_RECORD_WA, '|-') == 0) # Fork -- write to child
{
    &p_copy_copy_record_wa::copy_copy_record_wa;
    exit(0);
}

if (open(COPY_COPY_RECORD_SA, '|-') == 0) # Fork -- write to child
{
    &p_copy_copy_record_sa::copy_copy_record_sa;
    exit(0);
}

if (open(COPY_COPY_RECORD_NSW, '|-') == 0) # Fork -- write to child
{
    &p_copy_copy_record_nsw::copy_copy_record_nsw;
    exit(0);
}

if (open(COPY_COPY_RECORD_VIC, '|-') == 0) # Fork -- write to child
{
    &p_copy_copy_record_vic::copy_copy_record_vic;
    exit(0);
}

if (open(COPY_COPY_RECORD_NT, '|-') == 0) # Fork -- write to child
{
    &p_copy_copy_record_nt::copy_copy_record_nt;
    exit(0);
}

print STDERR "[examples/copy_record.pql ' . localtime() . "] Start";
```

```

use Benchmark;
my $benchmark_start = new Benchmark;
while (<CHAIN_PEQUEL_PTL>)
{
    print STDERR "[examples/copy_record.pql ' . localtime() . "] $. records." if ($. % VERBOSE == 0);
    chomp;
    @I_VAL = split("[|]", $_);
    next unless ($I_VAL[_I_LOCATION] eq 'WA' || $I_VAL[_I_LOCATION] eq 'SA' || $I_VAL[_I_LOCATION] eq 'NSW' ||
$I_VAL[_I_LOCATION] eq 'VIC' || $I_VAL[_I_LOCATION] eq 'NT');
    if ($I_VAL[_I_LOCATION] eq 'WA')
    {
        print COPY_COPY_RECORD_WA
            @I_VAL[0..LAST_ICELL];
    }

    if ($I_VAL[_I_LOCATION] eq 'SA')
    {
        print COPY_COPY_RECORD_SA
            @I_VAL[0..LAST_ICELL];
    }

    if ($I_VAL[_I_LOCATION] eq 'NSW')
    {
        print COPY_COPY_RECORD_NSW
            @I_VAL[0..LAST_ICELL];
    }

    if ($I_VAL[_I_LOCATION] eq 'VIC')
    {
        print COPY_COPY_RECORD_VIC
            @I_VAL[0..LAST_ICELL];
    }

    if ($I_VAL[_I_LOCATION] eq 'NT')
    {
        print COPY_COPY_RECORD_NT
            @I_VAL[0..LAST_ICELL];
    }

    $key__I_LOCATION = $I_VAL[_I_LOCATION];
    if (!defined($previous_key__I_LOCATION))
    {
        $previous_key__I_LOCATION = $key__I_LOCATION;
    }

    elsif ($previous_key__I_LOCATION ne $key__I_LOCATION)
    {
        flock(STDOUT, LOCK_EX);
        print STDOUT
            $O_VAL[_O_LOCATION],
            $O_VAL[_O_DESCRIPTION],
            $O_VAL[_O_SALES_TOTAL]
            ;
        flock(STDOUT, LOCK_UN);
        $previous_key__I_LOCATION = $key__I_LOCATION;
        @O_VAL = undef;
    }

    $I_VAL[_I_LOCATION_DESC] = $$TABLE_LOC_DESCRIPTOR{qq{$I_VAL[_I_LOCATION]}};
    $O_VAL[_O_LOCATION] = $I_VAL[_I_LOCATION_DESC];
    $I_VAL[_I_DESCRIPTION] = 'State Total';
    $O_VAL[_O_DESCRIPTION] = $I_VAL[_I_DESCRIPTION];
    $O_VAL[_O_SALES_TOTAL] += $I_VAL[_I_SALES_TOTAL] unless ($I_VAL[_I_SALES_TOTAL] eq '');
}

flock(STDOUT, LOCK_EX);
print STDOUT
    $O_VAL[_O_LOCATION],
    $O_VAL[_O_DESCRIPTION],
    $O_VAL[_O_SALES_TOTAL]
    ;
flock(STDOUT, LOCK_UN);
close(COPY_COPY_RECORD_NT);
close(COPY_COPY_RECORD_VIC);
close(COPY_COPY_RECORD_NSW);
close(COPY_COPY_RECORD_SA);
close(COPY_COPY_RECORD_WA);
close(STDOUT);
print STDERR "[examples/copy_record.pql ' . localtime() . "] $. records.";
my $benchmark_end = new Benchmark;
my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
print STDERR "[examples/copy_record.pql ' . localtime() . "] Code statistics: @[timestr($benchmark_timediff)]
";
#-----
##### Table LOC_DESCRIPTOR --> Type :Pequel::Type::Table::Local #####

```

```

sub InitLookupLOC_DESCRIPTOR
{
    my %_TABLE_LOC_DESCRIPTOR;
    %_TABLE_LOC_DESCRIPTOR =
    (
        'NSW' => 'New South Wales',
        'NT' => 'Northern Territory',
        'QLD' => 'Queensland',
        'SA' => 'South Australia',
        'VIC' => 'Victoria',
        'WA' => 'Western Australia'
    );
    return \%_TABLE_LOC_DESCRIPTOR;
}

{
    package p_execPequelCHAIN_PEQUEL_PT1;
    sub execPequelCHAIN_PEQUEL_PT1
    {
        vim: syntax=perl ts=4 sw=4
#-----
#   Generated By: pequel Version 2.3-4, Build: Wednesday October 12 23:16:49 BST 2005
#       : https://sourceforge.net/projects/pequel/
#   Script Name : examples/chain_pequel_pt1.pql
#   Created On  : Wed Oct 12 15:27:38 2005
#   For         :
#-----
#   Options:
#       input_file(sample.data) input data filename
#       optimize(1) optimize generated code.
#       hash(1) Generate in memory. Input data can be unsorted.
#       doc_title(Pequel Chaining Part-1 Example Script) document title.
#       doc_email(sample@youraddress.com) document email entry.
#       doc_version(2.3) 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 _O_LOCATION        => int    1;
        use constant _O_PRODUCT_CODE    => int    2;
        use constant _O_SALES_TOTAL     => int    3;
        local $\\="\n"; local $,="|";
        print STDERR '[examples/chain_pequel_pt1.pql ' . localtime() . "] Init";
        use constant VERBOSE => int 10000;
        use constant LAST_ICELL => int 8;
        my @I_VAL;
        my %O_VAL;
        my $key;
        open(DATA, q{examples/sample.data}) || die "Cannot open examples/sample.data: $!";
        open(STDOUT, '|-', q{sort -t'|' -y -k 1,1 2>/dev/null |});
        print STDERR '[examples/chain_pequel_pt1.pql ' . localtime() . "] Start";
        use Benchmark;
        my $benchmark_start = new Benchmark;
        while (<DATA>)
        {
            print STDERR '[examples/chain_pequel_pt1.pql ' . localtime() . "] $. records." if ($. % VERBOSE ==
0);
            chomp;
            @I_VAL = split("[|]", $_);
            $key = ( $I_VAL[_I_LOCATION] ) . '|' . ( $I_VAL[_I_PRODUCT_CODE] );
            $O_VAL{$key}{_O_LOCATION} = $I_VAL[_I_LOCATION];
            $O_VAL{$key}{_O_PRODUCT_CODE} = $I_VAL[_I_PRODUCT_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_LOCATION},
                $O_VAL{$key}{_O_PRODUCT_CODE},
                $O_VAL{$key}{_O_SALES_TOTAL}
            ;
        }

        close(STDOUT);
        print STDERR '[examples/chain_pequel_pt1.pql ' . localtime() . "] $. records.";
        my $benchmark_end = new Benchmark;

```

```

my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
print STDERR '[examples/chain_pequel_pt1.pql ' . localtime() . "] Code statistics: @{{timestr($benchmark_timediff)}}";
#-----
}

}

{
    package p_copy_copy_record_wa;
    sub copy_copy_record_wa
    {
        vim: syntax=perl ts=4 sw=4
#-----
#   Generated By: pequel Version 2.3-4, Build: Wednesday October 12 23:16:49 BST 2005
#               : https://sourceforge.net/projects/pequel/
#   Script Name : examples/copy_record_WA.pql
#   Created On  : Wed Oct 12 15:27:39 2005
#   For         :
#-----
#   Options:
#       optimize(1) optimize generated code.
#       doc_title(Copy Record Example Script) document title.
#       doc_email(sample@youraddress.com) document email entry.
#       doc_version(2.3) document version for pequel script.
#-----
        use strict;
        use Fcntl ':flock';
        use constant _I_LOCATION          => int    0;
        use constant _I_PRODUCT_CODE      => int    1;
        use constant _I_SALES_TOTAL       => int    2;
        use constant _I_LOCATION_NAME     => int    3;
        use constant _O_LOCATION_NAME     => int    1;
        use constant _O_PRODUCT_CODE      => int    2;
        use constant _O_SALES_TOTAL       => int    3;
        local $\\="\n"; local $,="|";
        print STDERR '[examples/copy_record_WA.pql ' . localtime() . "] Init";
        use constant VERBOSE => int 10000;
        use constant LAST_ICELL => int 3;
        my @I_VAL;
        my @O_VAL;
        my $key__I_PRODUCT_CODE;
        my $previous_key__I_PRODUCT_CODE = undef;
        foreach my $f (1..3) { $O_VAL[$f] = undef; }
#   Sort:PRODUCT_CODE(asc:string)
        open(DATA, q{cat - | sort -t'|' -y -k 2,2 2>/dev/null |}) || die "Cannot open input: $!";
        print STDERR '[examples/copy_record_WA.pql ' . localtime() . "] Start";
        use Benchmark;
        my $benchmark_start = new Benchmark;
        while (<DATA>)
        {
            print STDERR '[examples/copy_record_WA.pql ' . localtime() . "] $. records." if ($. % VERBOSE == 0);

            chomp;
            @I_VAL = split("[|]", $_);
            $key__I_PRODUCT_CODE = $I_VAL[_I_PRODUCT_CODE];
            if (!defined($previous_key__I_PRODUCT_CODE))
            {
                $previous_key__I_PRODUCT_CODE = $key__I_PRODUCT_CODE;
            }

            elsif ($previous_key__I_PRODUCT_CODE ne $key__I_PRODUCT_CODE)
            {
                flock(STDOUT, LOCK_EX);
                print STDOUT
                    $O_VAL[_O_LOCATION_NAME],
                    $O_VAL[_O_PRODUCT_CODE],
                    $O_VAL[_O_SALES_TOTAL]
                ;
                flock(STDOUT, LOCK_UN);
                $previous_key__I_PRODUCT_CODE = $key__I_PRODUCT_CODE;
                @O_VAL = undef;
            }

            $I_VAL[_I_LOCATION_NAME] = 'Western Australia';
            $O_VAL[_O_LOCATION_NAME] = $I_VAL[_I_LOCATION_NAME];
            $O_VAL[_O_PRODUCT_CODE] = $I_VAL[_I_PRODUCT_CODE];
            $O_VAL[_O_SALES_TOTAL] += $I_VAL[_I_SALES_TOTAL] unless ($I_VAL[_I_SALES_TOTAL] eq '');
        }

        flock(STDOUT, LOCK_EX);
        print STDOUT
            $O_VAL[_O_LOCATION_NAME],
            $O_VAL[_O_PRODUCT_CODE],
            $O_VAL[_O_SALES_TOTAL]

```

```

;
flock(STDOUT, LOCK_UN);
print STDERR '[examples/copy_record_WA.pql ' . localtime() . "] $. records.";
my $benchmark_end = new Benchmark;
my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
print STDERR '[examples/copy_record_WA.pql ' . localtime() . "] Code statistics: @{{timestr($benchmark
_timediff)}}";
#-----
}

{
    package p_copy_copy_record_sa;
    sub copy_copy_record_sa
    {
        # vim: syntax=perl ts=4 sw=4
#-----
#   Generated By: pequel Version 2.3-4, Build: Wednesday October 12 23:16:49 BST 2005
#   : https://sourceforge.net/projects/pequel/
#   Script Name : examples/copy_record_SA.pql
#   Created On  : Wed Oct 12 15:27:41 2005
#   For         :
#-----
#   Options:
#       optimize(1) optimize generated code.
#       doc_title(Copy Record Example Script) document title.
#       doc_email(sample@youraddress.com) document email entry.
#       doc_version(2.3) document version for pequel script.
#-----
        use strict;
        use Fcntl ':flock';
        use constant _I_LOCATION          => int    0;
        use constant _I_PRODUCT_CODE      => int    1;
        use constant _I_SALES_TOTAL       => int    2;
        use constant _I_LOCATION_NAME     => int    3;
        use constant _O_LOCATION_NAME     => int    1;
        use constant _O_PRODUCT_CODE      => int    2;
        use constant _O_SALES_TOTAL       => int    3;
        local $\="\n"; local $,="|";
        print STDERR '[examples/copy_record_SA.pql ' . localtime() . "] Init";
        use constant VERBOSE => int 10000;
        use constant LAST_ICELL => int 3;
        my @I_VAL;
        my @O_VAL;
        my $key__I_PRODUCT_CODE;
        my $previous_key__I_PRODUCT_CODE = undef;
        foreach my $f (1..3) { $O_VAL[$f] = undef; }
#   Sort:PRODUCT_CODE(asc:string)
        open(DATA, q{cat - | sort -t'|' -y -k 2,2 2>/dev/null |}) || die "Cannot open input: $!";
        print STDERR '[examples/copy_record_SA.pql ' . localtime() . "] Start";
        use Benchmark;
        my $benchmark_start = new Benchmark;
        while (<DATA>)
        {
            print STDERR '[examples/copy_record_SA.pql ' . localtime() . "] $. records." if ($. % VERBOSE == 0
);
            chomp;
            @I_VAL = split("[|]", $_);
            $key__I_PRODUCT_CODE = $I_VAL[_I_PRODUCT_CODE];
            if (!defined($previous_key__I_PRODUCT_CODE))
            {
                $previous_key__I_PRODUCT_CODE = $key__I_PRODUCT_CODE;
            }

            elsif ($previous_key__I_PRODUCT_CODE ne $key__I_PRODUCT_CODE)
            {
                flock(STDOUT, LOCK_EX);
                print STDOUT
                    $O_VAL[_O_LOCATION_NAME],
                    $O_VAL[_O_PRODUCT_CODE],
                    $O_VAL[_O_SALES_TOTAL]
                ;
                flock(STDOUT, LOCK_UN);
                $previous_key__I_PRODUCT_CODE = $key__I_PRODUCT_CODE;
                @O_VAL = undef;
            }

            $I_VAL[_I_LOCATION_NAME] = 'South Australia';
            $O_VAL[_O_LOCATION_NAME] = $I_VAL[_I_LOCATION_NAME];
            $O_VAL[_O_PRODUCT_CODE] = $I_VAL[_I_PRODUCT_CODE];
            $O_VAL[_O_SALES_TOTAL] += $I_VAL[_I_SALES_TOTAL] unless ($I_VAL[_I_SALES_TOTAL] eq '');
        }

        flock(STDOUT, LOCK_EX);

```

```

print STDOUT
    $O_VAL[_O_LOCATION_NAME],
    $O_VAL[_O_PRODUCT_CODE],
    $O_VAL[_O_SALES_TOTAL]
;
flock(STDOUT, LOCK_UN);
print STDERR '[examples/copy_record_SA.pql ' . localtime() . "] $. records.";
my $benchmark_end = new Benchmark;
my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
print STDERR '[examples/copy_record_SA.pql ' . localtime() . "] Code statistics: @{"timestr($benchmark
_timediff)"}";
#-----
}

}

{
    package p_copy_copy_record_nsw;
    sub copy_copy_record_nsw
    {
        vim: syntax=perl ts=4 sw=4
#-----
#   Generated By: pequel Version 2.3-4, Build: Wednesday October 12 23:16:49 BST 2005
#   : https://sourceforge.net/projects/pequel/
#   Script Name : examples/copy_record_NSW.pql
#   Created On  : Wed Oct 12 15:27:42 2005
#   For         :
#-----
#   Options:
#       optimize(1) optimize generated code.
#       doc_title(Copy Record Example Script) document title.
#       doc_email(sample@youraddress.com) document email entry.
#       doc_version(2.3) document version for pequel script.
#-----
        use strict;
        use Fcntl ':flock';
        use constant _I_LOCATION      => int    0;
        use constant _I_PRODUCT_CODE  => int    1;
        use constant _I_SALES_TOTAL   => int    2;
        use constant _I_LOCATION_NAME => int    3;
        use constant _O_LOCATION_NAME => int    1;
        use constant _O_PRODUCT_CODE  => int    2;
        use constant _O_SALES_TOTAL    => int    3;
        local $\\="\n"; local $,="|";
        print STDERR '[examples/copy_record_NSW.pql ' . localtime() . "] Init";
        use constant VERBOSE => int 10000;
        use constant LAST_ICELL => int 3;
        my @I_VAL;
        my @O_VAL;
        my $key__I_PRODUCT_CODE;
        my $previous_key__I_PRODUCT_CODE = undef;
        foreach my $f (1..3) { $O_VAL[$f] = undef; }
#   Sort:PRODUCT_CODE(asc:string)
        open(DATA, q{cat - | sort -t'|' -y -k 2,2 2>/dev/null |}) || die "Cannot open input: $!";
        print STDERR '[examples/copy_record_NSW.pql ' . localtime() . "] Start";
        use Benchmark;
        my $benchmark_start = new Benchmark;
        while (<DATA>)
        {
            print STDERR '[examples/copy_record_NSW.pql ' . localtime() . "] $. records." if ($. % VERBOSE ==
0);

            chomp;
            @I_VAL = split("[|]", $_);
            $key__I_PRODUCT_CODE = $I_VAL[_I_PRODUCT_CODE];
            if (!defined($previous_key__I_PRODUCT_CODE))
            {
                $previous_key__I_PRODUCT_CODE = $key__I_PRODUCT_CODE;
            }

            elsif ($previous_key__I_PRODUCT_CODE ne $key__I_PRODUCT_CODE)
            {
                flock(STDOUT, LOCK_EX);
                print STDOUT
                    $O_VAL[_O_LOCATION_NAME],
                    $O_VAL[_O_PRODUCT_CODE],
                    $O_VAL[_O_SALES_TOTAL]
                ;
                flock(STDOUT, LOCK_UN);
                $previous_key__I_PRODUCT_CODE = $key__I_PRODUCT_CODE;
                @O_VAL = undef;
            }

            $I_VAL[_I_LOCATION_NAME] = 'New South Wales';
            $O_VAL[_O_LOCATION_NAME] = $I_VAL[_I_LOCATION_NAME];
            $O_VAL[_O_PRODUCT_CODE] = $I_VAL[_I_PRODUCT_CODE];

```

```

        $O_VAL[_O_SALES_TOTAL] += $I_VAL[_I_SALES_TOTAL] unless ($I_VAL[_I_SALES_TOTAL] eq '');
    }

    flock(STDOUT, LOCK_EX);
    print STDOUT
        $O_VAL[_O_LOCATION_NAME],
        $O_VAL[_O_PRODUCT_CODE],
        $O_VAL[_O_SALES_TOTAL]
    ;
    flock(STDOUT, LOCK_UN);
    print STDERR '[examples/copy_record_NSW.pql ' . localtime() . "] $. records.";
    my $benchmark_end = new Benchmark;
    my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
    print STDERR '[examples/copy_record_NSW.pql ' . localtime() . "] Code statistics: @{" . timestr($benchmark_
k_timediff) . "}";
#-----
}

}

{
    package p_copy_copy_record_vic;
    sub copy_copy_record_vic
    {
        vim: syntax=perl ts=4 sw=4
#-----
#   Generated By: pequel Version 2.3-4, Build: Wednesday October 12 23:16:49 BST 2005
#               : https://sourceforge.net/projects/pequel/
#   Script Name : examples/copy_record_VIC.pql
#   Created On  : Wed Oct 12 15:27:43 2005
#   For         :
#-----
#   Options:
#       optimize(1) optimize generated code.
#       doc_title(Copy Record Example Script) document title.
#       doc_email(sample@youraddress.com) document email entry.
#       doc_version(2.3) document version for pequel script.
#-----
        use strict;
        use Fcntl ':flock';
        use constant _I_LOCATION          => int    0;
        use constant _I_PRODUCT_CODE      => int    1;
        use constant _I_SALES_TOTAL       => int    2;
        use constant _I_LOCATION_NAME     => int    3;
        use constant _O_LOCATION_NAME     => int    1;
        use constant _O_PRODUCT_CODE      => int    2;
        use constant _O_SALES_TOTAL       => int    3;
        local $\\="\n"; local $,="|";
        print STDERR '[examples/copy_record_VIC.pql ' . localtime() . "] Init";
        use constant VERBOSE => int 10000;
        use constant LAST_ICELL => int 3;
        my @I_VAL;
        my @O_VAL;
        my $key__I_PRODUCT_CODE;
        my $previous_key__I_PRODUCT_CODE = undef;
        foreach my $f (1..3) { $O_VAL[$f] = undef; }
#   Sort:PRODUCT_CODE(asc:string)
        open(DATA, q[cat - | sort -t'|' -y -k 2,2 2>/dev/null |]) || die "Cannot open input: $!";
        print STDERR '[examples/copy_record_VIC.pql ' . localtime() . "] Start";
        use Benchmark;
        my $benchmark_start = new Benchmark;
        while (<DATA>)
        {
            print STDERR '[examples/copy_record_VIC.pql ' . localtime() . "] $. records." if ($. % VERBOSE ==
0);
            chomp;
            @I_VAL = split("[|]", $_);
            $key__I_PRODUCT_CODE = $I_VAL[_I_PRODUCT_CODE];
            if (!defined($previous_key__I_PRODUCT_CODE))
            {
                $previous_key__I_PRODUCT_CODE = $key__I_PRODUCT_CODE;
            }

            elsif ($previous_key__I_PRODUCT_CODE ne $key__I_PRODUCT_CODE)
            {
                flock(STDOUT, LOCK_EX);
                print STDOUT
                    $O_VAL[_O_LOCATION_NAME],
                    $O_VAL[_O_PRODUCT_CODE],
                    $O_VAL[_O_SALES_TOTAL]
                ;
                flock(STDOUT, LOCK_UN);
                $previous_key__I_PRODUCT_CODE = $key__I_PRODUCT_CODE;
                @O_VAL = undef;
            }
        }
    }
}

```



```

    $I_VAL[_I_LOCATION_NAME] = 'Victoria';
    $O_VAL[_O_LOCATION_NAME] = $I_VAL[_I_LOCATION_NAME];
    $O_VAL[_O_PRODUCT_CODE] = $I_VAL[_I_PRODUCT_CODE];
    $O_VAL[_O_SALES_TOTAL] += $I_VAL[_I_SALES_TOTAL] unless ($I_VAL[_I_SALES_TOTAL] eq '');
}

flock(STDOUT, LOCK_EX);
print STDOUT
    $O_VAL[_O_LOCATION_NAME],
    $O_VAL[_O_PRODUCT_CODE],
    $O_VAL[_O_SALES_TOTAL]
;
flock(STDOUT, LOCK_UN);
print STDERR '[examples/copy_record_VIC.pql ' . localtime() . "] $. records.";
my $benchmark_end = new Benchmark;
my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
print STDERR '[examples/copy_record_VIC.pql ' . localtime() . "] Code statistics: @{{timestr($benchmark_timediff)}}";
#-----
}

{
    package p_copy_copy_record_nt;
    sub copy_copy_record_nt
    {
        # vim: syntax=perl ts=4 sw=4
        #-----
        # Generated By: pequel Version 2.3-4, Build: Wednesday October 12 23:16:49 BST 2005
        # : https://sourceforge.net/projects/pequel/
        # Script Name : examples/copy_record_NT.pql
        # Created On : Wed Oct 12 15:27:45 2005
        # For :
        #-----
        # Options:
        # optimize(1) optimize generated code.
        # doc_title(Copy Record Example Script) document title.
        # doc_email(sample@youraddress.com) document email entry.
        # doc_version(2.3) document version for pequel script.
        #-----
        use strict;
        use Fcntl ':flock';
        use constant _I_LOCATION => int 0;
        use constant _I_PRODUCT_CODE => int 1;
        use constant _I_SALES_TOTAL => int 2;
        use constant _I_LOCATION_NAME => int 3;
        use constant _O_LOCATION_NAME => int 1;
        use constant _O_PRODUCT_CODE => int 2;
        use constant _O_SALES_TOTAL => int 3;
        local $\\="\\n"; local $,="|";
        print STDERR '[examples/copy_record_NT.pql ' . localtime() . "] Init";
        use constant VERBOSE => int 10000;
        use constant LAST_ICELL => int 3;
        my @I_VAL;
        my @O_VAL;
        my $key__I_PRODUCT_CODE;
        my $previous_key__I_PRODUCT_CODE = undef;
        foreach my $f (1..3) { $O_VAL[$f] = undef; }
        # Sort:PRODUCT_CODE(asc:string)
        open(DATA, q{cat - | sort -t'|' -y -k 2,2 2>/dev/null |}) || die "Cannot open input: $!";
        print STDERR '[examples/copy_record_NT.pql ' . localtime() . "] Start";
        use Benchmark;
        my $benchmark_start = new Benchmark;
        while (<DATA>)
        {
            print STDERR '[examples/copy_record_NT.pql ' . localtime() . "] $. records." if ($. % VERBOSE == 0
);
            chomp;
            @I_VAL = split("[|]", $_);
            $key__I_PRODUCT_CODE = $I_VAL[_I_PRODUCT_CODE];
            if (!defined($previous_key__I_PRODUCT_CODE))
            {
                $previous_key__I_PRODUCT_CODE = $key__I_PRODUCT_CODE;
            }

            elsif ($previous_key__I_PRODUCT_CODE ne $key__I_PRODUCT_CODE)
            {
                flock(STDOUT, LOCK_EX);
                print STDOUT
                    $O_VAL[_O_LOCATION_NAME],
                    $O_VAL[_O_PRODUCT_CODE],
                    $O_VAL[_O_SALES_TOTAL]
                ;
                flock(STDOUT, LOCK_UN);

```

```

        $previous_key__I_PRODUCT_CODE = $key__I_PRODUCT_CODE;
        @O_VAL = undef;
    }

    $I_VAL[_I_LOCATION_NAME] = 'Northern Territory';
    $O_VAL[_O_LOCATION_NAME] = $I_VAL[_I_LOCATION_NAME];
    $O_VAL[_O_PRODUCT_CODE] = $I_VAL[_I_PRODUCT_CODE];
    $O_VAL[_O_SALES_TOTAL] += $I_VAL[_I_SALES_TOTAL] unless ($I_VAL[_I_SALES_TOTAL] eq '');
}

flock(STDOUT, LOCK_EX);
print STDOUT
    $O_VAL[_O_LOCATION_NAME],
    $O_VAL[_O_PRODUCT_CODE],
    $O_VAL[_O_SALES_TOTAL]
;
flock(STDOUT, LOCK_UN);
print STDERR '[examples/copy_record_NT.pql ' . localtime() . "] $. records.";
my $benchmark_end = new Benchmark;
my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
print STDERR '[examples/copy_record_NT.pql ' . localtime() . "] Code statistics: @{{timestr($benchmark
_timediff)}}";
#-----
}

}

```

## 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

