external_tables.pql
by *Pequel*

sample@youraddress.com

External Tables Example Script

2.2

Table of Contents
External Tables Example Script

SCRIPT NAME

external_tables.pql

DESCRIPTION

Demonstrates the use of external tables. The default method for loading an external table is to embed the table contents in the generated code. SAMPLE1 is a example of an embedded table. External tables may also be loaded dynamically (at runtime) — the '_' table name prefix instructs Pequel to load the table dynamically. SAMPLE2 is an axample of a dynamic table. The optional environment variable 'PEQUEL_TABLE_PATH' may be set to the path for the location of the table data-source-files. This path will be used to locate the data-source-files unless the data source filename is an absolute path name.

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

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

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

1.1 *PRODUCT_CODE*

Output Field

Description
Set to input field ***PRODUCT_CODE***

1.2 *RECORD_COUNT*

Output Field

Description
Count aggregation.

1.3 *SALES_QTY_SAMPLE1*

Output Field

Description
Sum aggregation on input field ***SALES_QTY***.

Aggregation condition
exists %SAMPLE1(PRODUCT_CODE);

1.4 *SALES_QTY_SAMPLE2*

Output Field

Description
Sum aggregation on input field ***SALES_QTY***.

Aggregation condition
exists %SAMPLE2(PRODUCT_CODE);

1.5 *S1_DESCRIPTION*

Output Field

Description

Set to input field ***S1_DESCRIPTION***

Derived Input Field Evaluation

```
=> %SAMPLE1(PRODUCT_CODE)->DESCRIPTION
```

1.6 *S1_LOCATION*

Output Field

Description

Set to input field ***S1_LOCATION***

Derived Input Field Evaluation

```
=> %SAMPLE1(PRODUCT_CODE)->LOCATION
```

1.7 *S2_DESCRIPTION*

Output Field

Description

Set to input field ***S2_DESCRIPTION***

Derived Input Field Evaluation

```
=> %SAMPLE2(PRODUCT_CODE)->DESCRIPTION
```

1.8 *S2_LOCATION*

Output Field

Description

Set to input field ***S2_LOCATION***

Derived Input Field Evaluation

```
=> %SAMPLE2(PRODUCT_CODE)->LOCATION
```

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: external_tables.pql

2.4 *header*

write header record to output.: 1

2.5 *optimize*

optimize generated code.: 1

2.6 *doc_title*

document title.: External Tables 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 *SAMPLE1*

Table Type: ***local***

Data

L100RWT02 — IBM 2000 IT P4-1600/512/40GB NT
K109KWO08 — Philips 8000 GR P3-1300/1GB/20GB NT
L106FEU09 — Toshiba 9000 EN P4-1900/1GB/60GB PER
B101XDR04 — HP 4000 SP P3-1400/128/60GB SA
L118DVB09 — Cannon 9000 FR P4-1600/256/30GB SA
J103CFG04 — Cannon 4000 EN P3-1400/128/20GB QLD
G113IVP03 — Philips 3000 GR P3-880/128/10GB WA
D104FNJ08 — Compaq 8000 GR P3-880/1GB/10GB VIC
E108HQQ06 — IBM 6000 SP P4-1800/512/60GB VIC
J108NQE07 — Dell 7000 IT P3-880/256/40GB QLD
G100YTF02 — IBM 2000 FR P4-1800/128/20GB QLD
L103ENU10 — Fujitsu 10000 GR P4-1800/128/10GB VIC
B103CUO10 — Toshiba 10000 SP P3-880/256/10GB WA
G105DJH10 — Fujitsu 10000 EN P3-880/1GB/30GB PER
G104QGG05 — Dell 5000 SP P3-1200/512/20GB WA
G116VHO02 — Dell 2000 SP P3-1200/128/30GB ALIC
C104LIS01 — Dell 1000 FR P4-1600/128/20GB SYD
E102ITQ08 — Philips 8000 EN P4-2000/512/30GB NSW
K102YAP03 — HP 3000 IT P3-1200/128/30GB NSW
G111IPN09 — Fujitsu 9000 FR P3-1200/1GB/40GB NSW

3.2 *SAMPLE2*

Table Type: ***external***
Data Source Filename: ***sample.data***
Key Field Number: ***1***

3.2.1 *DESCRIPTION = 3*

3.2.2 *LOCATION = 8*

4. TABLE INFORMATION SUMMARY

4.1 Table List Sorted By Table Name

SAMPLE1 — ***1*** (*local*)
SAMPLE2 — ***2*** (*external*)

5. EXTERNAL_TABLES.PQL

options

```
pequeldoc(pdf)
detail(1)
script_name(external_tables.pql)
header(1)
optimize(1)
doc_title(External Tables Example Script)
doc_email(sample@youraddress.com)
doc_version(2.2)
```

description

```
Demonstrates the use of external tables.
The default method for loading an external table is to embed the table contents in
the generated code. SAMPLE1 is a example of an embedded table. External tables
may also be loaded dynamically (at runtime) -- the '_' table name prefix instructs
Pequel to load the table dynamically. SAMPLE2 is an axample of a dynamic table.
The optional environment variable 'PEQUEL_TABLE_PATH' may be set to the path for
the location of the table data-source-files. This path will be used to locate the
data-source-files unless the data source filename is an absolute path name.
```

init table

```
SAMPLE1 L100RWT02 IBM 2000 IT P4-1600/512/40GB NT
SAMPLE1 K109KWO08 Philips 8000 GR P3-1300/1GB/20GB NT
SAMPLE1 L106FEU09 Toshiba 9000 EN P4-1900/1GB/60GB PER
SAMPLE1 B101XDR04 HP 4000 SP P3-1400/128/60GB SA
SAMPLE1 L118DVB09 Cannon 9000 FR P4-1600/256/30GB SA
SAMPLE1 J103CFG04 Cannon 4000 EN P3-1400/128/20GB QLD
SAMPLE1 G113IVP03 Philips 3000 GR P3-880/128/10GB WA
SAMPLE1 D104FNJ08 Compaq 8000 GR P3-880/1GB/10GB VIC
SAMPLE1 E108HQQ06 IBM 6000 SP P4-1800/512/60GB VIC
SAMPLE1 J108NQE07 Dell 7000 IT P3-880/256/40GB QLD
SAMPLE1 G100YTF02 IBM 2000 FR P4-1800/128/20GB QLD
SAMPLE1 L103ENU10 Fujitsu 10000 GR P4-1800/128/10GB VIC
SAMPLE1 B103CUO10 Toshiba 10000 SP P3-880/256/10GB WA
SAMPLE1 G105DJH10 Fujitsu 10000 EN P3-880/1GB/30GB PER
SAMPLE1 G104QGG05 Dell 5000 SP P3-1200/512/20GB WA
SAMPLE1 G116VHO02 Dell 2000 SP P3-1200/128/30GB ALIC
SAMPLE1 C104LIS01 Dell 1000 FR P4-1600/128/20GB SYD
SAMPLE1 E102ITQ08 Philips 8000 EN P4-2000/512/30GB NSW
SAMPLE1 K102YAP03 HP 3000 IT P3-1200/128/30GB NSW
SAMPLE1 G111IPN09 Fujitsu 9000 FR P3-1200/1GB/40GB NSW
```

load table

```
SAMPLE1 /* Table Name */ \
    sample.data /* Data Source Filename */ \
    1 /* Key Column Number */ \
     \
    DESCRIPTION = 3 \
    LOCATION = 8

SAMPLE2 /* Table Name */ \
    sample.data /* Data Source Filename */ \
    1 /* Key Column Number */ \
     \
    DESCRIPTION = 3 \
    LOCATION = 8
```

sort by

```
PRODUCT_CODE string
```

group by

```
PRODUCT_CODE string
```

input section

```
PRODUCT_CODE
COST_PRICE
DESCRIPTION
SALES_CODE
SALES_PRICE
```

```
SALES_QTY
SALES_DATE
LOCATION
S1_DESCRIPTION => %SAMPLE1(PRODUCT_CODE)->DESCRIPTION

S1_LOCATION => %SAMPLE1(PRODUCT_CODE)->LOCATION

S2_DESCRIPTION => %SAMPLE2(PRODUCT_CODE)->DESCRIPTION

S2_LOCATION => %SAMPLE2(PRODUCT_CODE)->LOCATION
```

output section

```
string    PRODUCT_CODE      PRODUCT_CODE
numeric   RECORD_COUNT      count *
numeric   SALES_QTY_SAMPLE1 sum SALES_QTY where exists %SAMPLE1(PRODUCT_CODE)
numeric   SALES_QTY_SAMPLE2 sum SALES_QTY where exists %SAMPLE2(PRODUCT_CODE)
string    S1_DESCRIPTION    S1_DESCRIPTION
string    S1_LOCATION       S1_LOCATION
string    S2_DESCRIPTION    S2_DESCRIPTION
string    S2_LOCATION       S2_LOCATION
```

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 : external_tables.pql
#Created On  : Tue Sep 13 10:19:23 2005
#For         :
#-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
#Options:
#pequeldoc(pdf) generate pod / pdf pequel script Reference Guide.
#detail(1) Include Pequel Generated Program chapter in Pequeldoc
#script_name(external_tables.pql) script filename
#header(1) write header record to output.
#optimize(1) optimize generated code.
#doc_title(External Tables 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 ’[external_tables.pql ’ . localtime() . "] Init";
use constant VERBOSE => int 10000;
use constant LAST_ICELL => int 11;
my @I_VAL;
my @O_VAL;
my $key__I_PRODUCT_CODE;
my $previous_key__I_PRODUCT_CODE = undef;
foreach my $f (1..8) { $O_VAL[$f] = undef; }
my $_TABLE_SAMPLE1 = &InitLookupSAMPLE1; # ref to %$SAMPLE1 hash
my $_TABLE_SAMPLE2 = &LoadTableSAMPLE2; # ref to %$SAMPLE2 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_S1_DESCRIPTION       => int    8;
use constant _I_S1_LOCATION          => int    9;
use constant _I_S2_DESCRIPTION       => int   10;
use constant _I_S2_LOCATION          => int   11;
use constant _O_PRODUCT_CODE         => int    1;
use constant _O_RECORD_COUNT         => int    2;
use constant _O_SALES_QTY_SAMPLE1    => int    3;
use constant _O_SALES_QTY_SAMPLE2    => int    4;
use constant _O_S1_DESCRIPTION       => int    5;
use constant _O_S1_LOCATION          => int    6;
use constant _O_S2_DESCRIPTION       => int    7;
use constant _O_S2_LOCATION          => int    8;
use constant _T_SAMPLE1_FLD_DESCRIPTION   => int    0;
use constant _T_SAMPLE1_FLD_LOCATION      => int    1;
use constant _T_SAMPLE2_FLD_DESCRIPTION   => int    0;
use constant _T_SAMPLE2_FLD_LOCATION      => int    1;
use constant _I_SAMPLE1_PRODUCT_CODE_FLD_KEY          => int   12;
use constant _I_SAMPLE1_PRODUCT_CODE_FLD_DESCRIPTION  => int   13;
use constant _I_SAMPLE1_PRODUCT_CODE_FLD_LOCATION     => int   14;
use constant _I_SAMPLE2_PRODUCT_CODE_FLD_KEY          => int   15;
use constant _I_SAMPLE2_PRODUCT_CODE_FLD_DESCRIPTION  => int   16;
use constant _I_SAMPLE2_PRODUCT_CODE_FLD_LOCATION     => int   17;
open(DATA, q{cat - | sort -t’|’ -y -k 1,1 |}) || die "Cannot open input: $!";
&PrintHeader();
print STDERR ’[external_tables.pql ’ . localtime() . "] Start";
use Benchmark;
my $benchmark_start = new Benchmark;
while (<DATA>)
{
    print STDERR ’[external_tables.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)
    {
        print
            $O_VAL[_O_PRODUCT_CODE],
            $O_VAL[_O_RECORD_COUNT],
            $O_VAL[_O_SALES_QTY_SAMPLE1],
            $O_VAL[_O_SALES_QTY_SAMPLE2],
            $O_VAL[_O_S1_DESCRIPTION],
            $O_VAL[_O_S1_LOCATION],
            $O_VAL[_O_S2_DESCRIPTION],
            $O_VAL[_O_S2_LOCATION]
        ;
        $previous_key__I_PRODUCT_CODE = $key__I_PRODUCT_CODE;
        @O_VAL = undef;
    }

    $O_VAL[_O_PRODUCT_CODE] = $I_VAL[_I_PRODUCT_CODE];
    $O_VAL[_O_RECORD_COUNT]++;
    $I_VAL[_I_S1_DESCRIPTION] = ${$$_TABLE_SAMPLE1{qq{$I_VAL[_I_PRODUCT_CODE]}}}[_T_SAMPLE1_FLD_DESCRIPTION];
    $O_VAL[_O_S1_DESCRIPTION] = $I_VAL[_I_S1_DESCRIPTION];
    $I_VAL[_I_S1_LOCATION] = ${$$_TABLE_SAMPLE1{qq{$I_VAL[_I_PRODUCT_CODE]}}}[_T_SAMPLE1_FLD_LOCATION];
    $O_VAL[_O_S1_LOCATION] = $I_VAL[_I_S1_LOCATION];
    $I_VAL[_I_S2_DESCRIPTION] = ${$$_TABLE_SAMPLE2{qq{$I_VAL[_I_PRODUCT_CODE]}}}[_T_SAMPLE2_FLD_DESCRIPTION];
    $O_VAL[_O_S2_DESCRIPTION] = $I_VAL[_I_S2_DESCRIPTION];
    $I_VAL[_I_S2_LOCATION] = ${$$_TABLE_SAMPLE2{qq{$I_VAL[_I_PRODUCT_CODE]}}}[_T_SAMPLE2_FLD_LOCATION];
    $O_VAL[_O_S2_LOCATION] = $I_VAL[_I_S2_LOCATION];

    if (exists $$_TABLE_SAMPLE1{qq{$I_VAL[_I_PRODUCT_CODE]}}) {
        $O_VAL[_O_SALES_QTY_SAMPLE1] += $I_VAL[_I_SALES_QTY] unless ($I_VAL[_I_SALES_QTY] eq '');
    }

    if (exists $$_TABLE_SAMPLE2{qq{$I_VAL[_I_PRODUCT_CODE]}}) {
        $O_VAL[_O_SALES_QTY_SAMPLE2] += $I_VAL[_I_SALES_QTY] unless ($I_VAL[_I_SALES_QTY] eq '');
    }
}

print
    $O_VAL[_O_PRODUCT_CODE],
    $O_VAL[_O_RECORD_COUNT],
    $O_VAL[_O_SALES_QTY_SAMPLE1],
    $O_VAL[_O_SALES_QTY_SAMPLE2],
    $O_VAL[_O_S1_DESCRIPTION],
    $O_VAL[_O_S1_LOCATION],
    $O_VAL[_O_S2_DESCRIPTION],
    $O_VAL[_O_S2_LOCATION]
;
print STDERR '[external_tables.pql ' . localtime() . "] $. records.";
my $benchmark_end = new Benchmark;
my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
print STDERR '[external_tables.pql ' . localtime() . "] Code statistics: @{[timestr($benchmark_timediff)]}";
#-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
#++++++ Table SAMPLE1 --> Type :Pequel::Type::Table::Local ++++++
sub InitLookupSAMPLE1
{
    my %_TABLE_SAMPLE1;
    %_TABLE_SAMPLE1 =
    (
        'B101XDR04' => ['HP 4000 SP P3-1400/128/60GB', 'SA'],
        'B103CUO10' => ['Toshiba 10000 SP P3-880/256/10GB', 'WA'],
        'C104LIS01' => ['Dell 1000 FR P4-1600/128/20GB', 'SYD'],
        'D104FNJ08' => ['Compaq 8000 GR P3-880/1GB/10GB', 'VIC'],
        'E102ITQ08' => ['Philips 8000 EN P4-2000/512/30GB', 'NSW'],
        'E108HQQ06' => ['IBM 6000 SP P4-1800/512/60GB', 'VIC'],
        'G100YTF02' => ['IBM 2000 FR P4-1800/128/20GB', 'QLD'],
        'G104QGG05' => ['Dell 5000 SP P3-1200/512/20GB', 'WA'],
        'G105DJH10' => ['Fujitsu 10000 EN P3-880/1GB/30GB', 'PER'],
        'G111IPN09' => ['Fujitsu 9000 FR P3-1200/1GB/40GB', 'NSW'],
        'G113IVP03' => ['Philips 3000 GR P3-880/128/10GB', 'WA'],
        'G116VHO02' => ['Dell 2000 SP P3-1200/128/30GB', 'ALIC'],
        'J103CFG04' => ['Cannon 4000 EN P3-1400/128/20GB', 'QLD'],
        'J108NQE07' => ['Dell 7000 IT P3-880/256/40GB', 'QLD'],
        'K102YAP03' => ['HP 3000 IT P3-1200/128/30GB', 'NSW'],
        'K109KWO08' => ['Philips 8000 GR P3-1300/1GB/20GB', 'NT'],
        'L100RWT02' => ['IBM 2000 IT P4-1600/512/40GB', 'NT'],
        'L103ENU10' => ['Fujitsu 10000 GR P4-1800/128/10GB', 'VIC'],
        'L106FEU09' => ['Toshiba 9000 EN P4-1900/1GB/60GB', 'PER'],
        'L118DVB09' => ['Cannon 9000 FR P4-1600/256/30GB', 'SA']
    );
    return \%_TABLE_SAMPLE1;
}

#++++++ Table SAMPLE2 --> Type :Pequel::Type::Table::External ++++++
sub LoadTableSAMPLE2
{
    my %_TABLE_SAMPLE2;
    my $dsf = "@{[ $ENV{PEQUEL_TABLE_PATH} eq '' ? '' : $ENV{PEQUEL_TABLE_PATH} . '/']}" . 'sample.data';
    print STDERR '[external_tables.pql ' . localtime() . "] Loading table SAMPLE2 from $dsf...";
    open(SAMPLE2, "sort -u -t'|' -k 1 $dsf |") || die("Unable to open table source file $dsf");
    while (<SAMPLE2>)
    {
        chomp;
        my (@flds) = split("[|]", $_, -1);
        $_TABLE_SAMPLE2{$flds[0]} = [ @flds[ 2,7 ]];
        print STDERR '[external_tables.pql ' . localtime() . "] Table SAMPLE2 $. records..." if ($. % 100000 =
= 0);
    }

    print STDERR '[external_tables.pql ' . localtime() . "] Table SAMPLE2 loaded $. records.";
    close(SAMPLE2);
    return \%_TABLE_SAMPLE2;
}

sub PrintHeader
{
    local $\="\n";
    local $,="|";
    print
        'PRODUCT_CODE',
        'RECORD_COUNT',
        'SALES_QTY_SAMPLE1',
        'SALES_QTY_SAMPLE2',
        'S1_DESCRIPTION',
        'S1_LOCATION',
        'S2_DESCRIPTION',
        'S2_LOCATION'
        ;
}
```

7. ABOUT PEQUEL

This document was generated by Pequel.

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

COPYRIGHT