



examples/external_tables.pql

by *Pequel*

sample@youraddress.com

External Tables Example Script

2.2

Table of Contents

External Tables Example Script

SCRIPT NAME	1
DESCRIPTION	1
1. PROCESS DETAILS	1
1.1 PRODUCT_CODE	1
Description	1
1.2 RECORD_COUNT	1
Description	1
1.3 SALES_QTY_SAMPLE1	1
Description	1
Aggregation condition	1
1.4 SALES_QTY_SAMPLE2	1
Description	1
Aggregation condition	1
1.5 S1_DESCRIPTION	2
Description	2
Derived Input Field Evaluation	2
1.6 S1_LOCATION	2
Description	2
Derived Input Field Evaluation	2
1.7 S2_DESCRIPTION	2
Description	2
Derived Input Field Evaluation	2
1.8 S2_LOCATION	2
Description	2
Derived Input 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 doc_title	3
2.8 doc_email	3
2.9 doc_version	3
3. TABLES	4
3.1 SAMPLE1	4
Data	4
3.2 SAMPLE2	5
4. TABLE INFORMATION SUMMARY	6
4.1 Table List Sorted By Table Name	6
5. EXAMPLES/EXTERNAL_TABLES.PQL	7
options	7
description	7
init table	7
load table	8
input section	8
sort by	8
group by	8
output section	8
6. PEQUEL GENERATED PROGRAM	9
7. ABOUT PEQUEL	12
COPYRIGHT	12

SCRIPT NAME

examples/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

DescriptionSet to input field **S1_DESCRIPTION****Derived Input Field Evaluation**

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

1.6 S1_LOCATION

Output Field

DescriptionSet to input field **S1_LOCATION****Derived Input Field Evaluation**

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

1.7 S2_DESCRIPTION

Output Field

DescriptionSet to input field **S2_DESCRIPTION****Derived Input Field Evaluation**

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

1.8 S2_LOCATION

Output Field

DescriptionSet to input field **S2_LOCATION****Derived Input Field Evaluation**

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

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/external_tables.pql

2.5 *header*

write header record to output.: 1

2.6 *optimize*

optimize generated code.: 1

2.7 *doc_title*

document title.: External Tables Example Script

2.8 *doc_email*

document email entry.: sample@youraddress.com

2.9 *doc_version*

document version for pequel script.: 2.2

3. TABLES

3.1 SAMPLE1

Table Type: *local*

Data

L103BJG04 — Toshiba 4000 IT P4-1800/1GB/60GB WA
A100AIX09 — Compaq 9000 GR P4-1700/256/40GB WA
B111KYK01 — Dell 1000 FR P4-1700/128/40GB PER
E100QTG07 — Fujitsu 7000 SP P4-1700/512/10GB NT
K113JAD05 — Fujitsu 5000 IT P3-1200/512/10GB PER
J115JBW09 — Compaq 9000 IT P3-1200/128/40GB SYD
J109NYP03 — HP 3000 IT P3-880/128/10GB MEL
A106UIH04 — Toshiba 4000 GR P4-1700/256/40GB ALIC
H107VAE06 — Toshiba 6000 FR P3-880/512/20GB WA
F104ICW08 — Compaq 8000 SP P4-1700/128/60GB PER
C103WEO02 — Cannon 2000 FR P4-1600/128/60GB WA
I108THJ06 — Dell 6000 GR P3-880/128/40GB VIC
D105BWE02 — IBM 2000 IT P4-1700/1GB/60GB PER
G111FOI06 — Toshiba 6000 FR P4-1900/512/60GB NT
I111AGN09 — Toshiba 9000 GR P4-1700/256/10GB PER
J102MLC05 — Fujitsu 5000 IT P3-1200/1GB/60GB VIC
G113WVH04 — Compaq 4000 SP P4-1800/256/20GB NT
I109JTE07 — IBM 7000 GR P3-1200/512/40GB MEL
C119GHQ10 — Dell 10000 FR P4-1700/1GB/30GB SYD
I115YVQ02 — Cannon 2000 EN P4-2000/256/10GB NSW
F105RTJ10 — Dell 10000 FR P3-900/512/20GB WA
A109IWD09 — Compaq 9000 IT P4-1700/128/20GB QLD
E119HQG01 — Dell 1000 GR P4-2000/1GB/40GB NT
A112HHM10 — Cannon 10000 FR P3-880/256/30GB SYD
K112WIS07 — Dell 7000 IT P3-1200/256/20GB PER
J112YXH07 — IBM 7000 EN P3-1400/256/40GB VIC
I105RHR09 — IBM 9000 FR P3-1200/512/40GB NT
L116RWV08 — Philips 8000 SP P3-900/128/10GB NSW
D117WMU02 — HP 2000 GR P4-1800/1GB/20GB QLD
C119HJM01 — Philips 1000 IT P3-1400/512/40GB NSW
L118PFA09 — Philips 9000 IT P4-1800/128/30GB SYD
E112SJD07 — IBM 7000 GR P3-1200/1GB/20GB SYD
F102EUR03 — Cannon 3000 EN P4-2000/512/30GB MEL
B117DAR07 — Cannon 7000 SP P4-1800/128/40GB ALIC
G103TKH08 — Fujitsu 8000 SP P4-1700/128/60GB ALIC
G106VOK04 — Fujitsu 4000 SP P3-900/512/40GB NT
F117WIP08 — IBM 8000 IT P3-900/1GB/10GB MEL
L105HMB07 — Philips 7000 FR P4-1600/1GB/10GB MEL
H113KDM07 — Compaq 7000 EN P3-880/512/40GB NT
C114ERT05 — IBM 5000 IT P4-1800/1GB/30GB VIC
H106LAF10 — Dell 10000 GR P4-2000/1GB/40GB SA
E100JMA04 — Cannon 4000 FR P3-1200/512/10GB VIC
E104HDH01 — Compaq 1000 EN P3-1200/256/20GB QLD
A109AYU10 — IBM 10000 FR P4-1700/512/10GB MEL
K111HOR02 — Cannon 2000 EN P4-1700/128/20GB NT
J112XUI05 — Dell 5000 EN P3-880/512/30GB PER
J117YTJ03 — IBM 3000 EN P4-1900/128/20GB VIC
D113QFU10 — Compaq 10000 SP P4-1900/1GB/30GB WA
K106NSX06 — Fujitsu 6000 IT P3-900/256/20GB NT
E108UFJ05 — Compaq 5000 SP P3-880/128/30GB VIC

3.2 SAMPLE2Table 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. EXAMPLES/EXTERNAL_TABLES.PQL

options

```

prefix(examples)
pequeldoc(pdf)
detail(1)
script_name(examples/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 an 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 example 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 L103BJG04 Toshiba 4000 IT P4-1800/1GB/60GB WA
SAMPLE1 A100AIX09 Compaq 9000 GR P4-1700/256/40GB WA
SAMPLE1 B111KYK01 Dell 1000 FR P4-1700/128/40GB PER
SAMPLE1 E100QTG07 Fujitsu 7000 SP P4-1700/512/10GB NT
SAMPLE1 K113JAD05 Fujitsu 5000 IT P3-1200/512/10GB PER
SAMPLE1 J115JBW09 Compaq 9000 IT P3-1200/128/40GB SYD
SAMPLE1 J109NYP03 HP 3000 IT P3-880/128/10GB MEL
SAMPLE1 A106UIH04 Toshiba 4000 GR P4-1700/256/40GB ALIC
SAMPLE1 H107VAE06 Toshiba 6000 FR P3-880/512/20GB WA
SAMPLE1 F104ICW08 Compaq 8000 SP P4-1700/128/60GB PER
SAMPLE1 C103WEO02 Cannon 2000 FR P4-1600/128/60GB WA
SAMPLE1 I108THJ06 Dell 6000 GR P3-880/128/40GB VIC
SAMPLE1 D105BWE02 IBM 2000 IT P4-1700/1GB/60GB PER
SAMPLE1 G111FOI06 Toshiba 6000 FR P4-1900/512/60GB NT
SAMPLE1 I111AGN09 Toshiba 9000 GR P4-1700/256/10GB PER
SAMPLE1 J102MLC05 Fujitsu 5000 IT P3-1200/1GB/60GB VIC
SAMPLE1 G113WVH04 Compaq 4000 SP P4-1800/256/20GB NT
SAMPLE1 I109JTE07 IBM 7000 GR P3-1200/512/40GB MEL
SAMPLE1 C119GHQ10 Dell 10000 FR P4-1700/1GB/30GB SYD
SAMPLE1 I115YVQ02 Cannon 2000 EN P4-2000/256/10GB NSW
SAMPLE1 F105RTJ10 Dell 10000 FR P3-900/512/20GB WA
SAMPLE1 A109IWD09 Compaq 9000 IT P4-1700/128/20GB QLD
SAMPLE1 E119HQG01 Dell 1000 GR P4-2000/1GB/40GB NT
SAMPLE1 A112HHM10 Cannon 10000 FR P3-880/256/30GB SYD
SAMPLE1 K112WIS07 Dell 7000 IT P3-1200/256/20GB PER
SAMPLE1 J112YXH07 IBM 7000 EN P3-1400/256/40GB VIC
SAMPLE1 I105RHR09 IBM 9000 FR P3-1200/512/40GB NT
SAMPLE1 L116RWV08 Philips 8000 SP P3-900/128/10GB NSW
SAMPLE1 D117WMU02 HP 2000 GR P4-1800/1GB/20GB QLD
SAMPLE1 C119HJM01 Philips 1000 IT P3-1400/512/40GB NSW
SAMPLE1 L118PFA09 Philips 9000 IT P4-1800/128/30GB SYD
SAMPLE1 E112SJD07 IBM 7000 GR P3-1200/1GB/20GB SYD
SAMPLE1 F102EUR03 Cannon 3000 EN P4-2000/512/30GB MEL
SAMPLE1 B117DAR07 Cannon 7000 SP P4-1800/128/40GB ALIC
SAMPLE1 G103TKH08 Fujitsu 8000 SP P4-1700/128/60GB ALIC
SAMPLE1 G106VOK04 Fujitsu 4000 SP P3-900/512/40GB NT
SAMPLE1 F117WIP08 IBM 8000 IT P3-900/1GB/10GB MEL
SAMPLE1 L105HMB07 Philips 7000 FR P4-1600/1GB/10GB MEL
SAMPLE1 H113KDM07 Compaq 7000 EN P3-880/512/40GB NT
SAMPLE1 C114ERT05 IBM 5000 IT P4-1800/1GB/30GB VIC
SAMPLE1 H106LAF10 Dell 10000 GR P4-2000/1GB/40GB SA
SAMPLE1 E100JMA04 Cannon 4000 FR P3-1200/512/10GB VIC
SAMPLE1 E104HDH01 Compaq 1000 EN P3-1200/256/20GB QLD
SAMPLE1 A109AYU10 IBM 10000 FR P4-1700/512/10GB MEL
SAMPLE1 K111HOR02 Cannon 2000 EN P4-1700/128/20GB NT
SAMPLE1 J112XUI05 Dell 5000 EN P3-880/512/30GB PER
SAMPLE1 J117YTJ03 IBM 3000 EN P4-1900/128/20GB VIC
SAMPLE1 D113QFU10 Compaq 10000 SP P4-1900/1GB/30GB WA
SAMPLE1 K106NSX06 Fujitsu 6000 IT P3-900/256/20GB NT
SAMPLE1 E108UFJ05 Compaq 5000 SP P3-880/128/30GB VIC

```

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

```

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

```

sort by

```
PRODUCT_CODE string
```

group by

```
PRODUCT_CODE string
```

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

```
#!/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 : external_tables.pql
#Created On  : Wed Nov 16 14:03:24 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/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;
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;
local $\\="\n";
local $,="|";
print STDERR "[examples/external_tables.pql ' . localtime() . "] Init";
use constant VERBOSE => int 10000;
use constant LAST_ICELL => int 11;
my @I_VAL;
my @O_VAL;
my $_inprec=0;
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
# Sort:PRODUCT_CODE(asc:string)
open(DATA, q{cat - | sort -t'|' -y -k 1,1 2>/dev/null |}) || die "Cannot open input: $!";
&PrintHeader();
print STDERR "[examples/external_tables.pql ' . localtime() . "] Start";
use Benchmark;
my $benchmark_start = new Benchmark;
while (<DATA>)
{
    ++$_inprec;
    print STDERR "[examples/external_tables.pql ' . localtime() . "] $_inprec records." if ($_inprec % VERBO
SE == 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 STDOUT
        $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 STDOUT
    $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]
;
close(DATA);
print STDERR '[examples/external_tables.pql ' . localtime() . "] $_inprecs records.";
my $benchmark_end = new Benchmark;
my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
print STDERR '[examples/external_tables.pql ' . localtime() . "] Code statistics: @{{timestr($benchmark_timediff)}}";
#-+-+-+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
#+++++ Table SAMPLE1 --> Type :ETL::Pequel::Type::Table::Local +++++
sub InitLookupSAMPLE1
{
    my %_TABLE_SAMPLE1;
    %_TABLE_SAMPLE1 =
    (
        'A100AIX09' => ['Compaq 9000 GR P4-1700/256/40GB', 'WA'],
        'A106UIH04' => ['Toshiba 4000 GR P4-1700/256/40GB', 'ALIC'],
        'A109AYU10' => ['IBM 10000 FR P4-1700/512/10GB', 'MEL'],
        'A109IWD09' => ['Compaq 9000 IT P4-1700/128/20GB', 'QLD'],
        'A112HHM10' => ['Cannon 10000 FR P3-880/256/30GB', 'SYD'],
        'B111KYK01' => ['Dell 1000 FR P4-1700/128/40GB', 'PER'],
        'B117DAR07' => ['Cannon 7000 SP P4-1800/128/40GB', 'ALIC'],
        'C103WEO02' => ['Cannon 2000 FR P4-1600/128/60GB', 'WA'],
        'C114ERT05' => ['IBM 5000 IT P4-1800/1GB/30GB', 'VIC'],
        'C119GHQ10' => ['Dell 10000 FR P4-1700/1GB/30GB', 'SYD'],
        'C119HJM01' => ['Philips 1000 IT P3-1400/512/40GB', 'NSW'],
        'D105BWE02' => ['IBM 2000 IT P4-1700/1GB/60GB', 'PER'],
        'D113QFU10' => ['Compaq 10000 SP P4-1900/1GB/30GB', 'WA'],
        'D117WMU02' => ['HP 2000 GR P4-1800/1GB/20GB', 'QLD'],
        'E100JMA04' => ['Cannon 4000 FR P3-1200/512/10GB', 'VIC'],
        'E100QTG07' => ['Fujitsu 7000 SP P4-1700/512/10GB', 'NT'],
        'E104HDH01' => ['Compaq 1000 EN P3-1200/256/20GB', 'QLD'],
        'E108UFJ05' => ['Compaq 5000 SP P3-880/128/30GB', 'VIC'],
        'E112SJD07' => ['IBM 7000 GR P3-1200/1GB/20GB', 'SYD'],
        'E119HQQ01' => ['Dell 1000 GR P4-2000/1GB/40GB', 'NT'],
        'F102EUR03' => ['Cannon 3000 EN P4-2000/512/30GB', 'MEL'],
    )
}

```

```

'F104ICW08' => ['Compaq 8000 SP P4-1700/128/60GB', 'PER'],
'F105RTJ10' => ['Dell 10000 FR P3-900/512/20GB', 'WA'],
'F117WIP08' => ['IBM 8000 IT P3-900/1GB/10GB', 'MEL'],
'G103TKH08' => ['Fujitsu 8000 SP P4-1700/128/60GB', 'ALIC'],
'G106VOK04' => ['Fujitsu 4000 SP P3-900/512/40GB', 'NT'],
'G111FOI06' => ['Toshiba 6000 FR P4-1900/512/60GB', 'NT'],
'G113WVH04' => ['Compaq 4000 SP P4-1800/256/20GB', 'NT'],
'H106LAF10' => ['Dell 10000 GR P4-2000/1GB/40GB', 'SA'],
'H107VAE06' => ['Toshiba 6000 FR P3-880/512/20GB', 'WA'],
'H113KDM07' => ['Compaq 7000 EN P3-880/512/40GB', 'NT'],
'I105RHR09' => ['IBM 9000 FR P3-1200/512/40GB', 'NT'],
'I108THJ06' => ['Dell 6000 GR P3-880/128/40GB', 'VIC'],
'I109JTE07' => ['IBM 7000 GR P3-1200/512/40GB', 'MEL'],
'I111AGN09' => ['Toshiba 9000 GR P4-1700/256/10GB', 'PER'],
'I115YVQ02' => ['Cannon 2000 EN P4-2000/256/10GB', 'NSW'],
'J102MLC05' => ['Fujitsu 5000 IT P3-1200/1GB/60GB', 'VIC'],
'J109NYP03' => ['HP 3000 IT P3-880/128/10GB', 'MEL'],
'J112XUI05' => ['Dell 5000 EN P3-880/512/30GB', 'PER'],
'J112XXH07' => ['IBM 7000 EN P3-1400/256/40GB', 'VIC'],
'J115JBW09' => ['Compaq 9000 IT P3-1200/128/40GB', 'SYD'],
'J117TJ03' => ['IBM 3000 EN P4-1900/128/20GB', 'VIC'],
'K106NSX06' => ['Fujitsu 6000 IT P3-900/256/20GB', 'NT'],
'K111HOR02' => ['Cannon 2000 EN P4-1700/128/20GB', 'NT'],
'K112WIS07' => ['Dell 7000 IT P3-1200/256/20GB', 'PER'],
'K113JAD05' => ['Fujitsu 5000 IT P3-1200/512/10GB', 'PER'],
'L103BJG04' => ['Toshiba 4000 IT P4-1800/1GB/60GB', 'WA'],
'L105HMB07' => ['Philips 7000 FR P4-1600/1GB/10GB', 'MEL'],
'L116RWV08' => ['Philips 8000 SP P3-900/128/10GB', 'NSW'],
'L118PFA09' => ['Philips 9000 IT P4-1800/128/30GB', 'SYD']

);
return \%_TABLE_SAMPLE1;
}

##### Table SAMPLE2 --> Type :ETL::Pequel::Type::Table::External #####
sub LoadTableSAMPLE2
{
    my \%_TABLE_SAMPLE2;
    my $dsf = 'examples/sample.data';
    print STDERR "[examples/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 "[examples/external_tables.pql ' . localtime() . "] Table SAMPLE2 $. records..." if ($. %
100000 == 0);
    }

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

sub PrintHeader
{
    local $\\n = "\n";
    local $, = "|";
    print STDOUT
        '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

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

