cornerstone

PARSER CODE

```
CREATE OR REPLACE PACKAGE plsql_parser AS
-------------------------------------------------------------
-- Written By  : David Rincon
--               http://www.cdsiusa.com
--               Copyright Cornerstone Data Systems, INC
-------------------------------------------------------------
-------------------------------------------------------------
-- Description: CIS procedures and functions
--
-------------------------------------------------------------
-- Modification History
--
-- Who          When          What
-- ------------ ------------- --------------------------------
-- D. Rincon    05/15/15      Created
-------------------------------------------------------------

   -------------------------------------------------------------
   -- SW Short String / LW Long Word / ST Short Text /
   -- LT Long Text
   -------------------------------------------------------------
   SS       CONSTANT NUMBER := 35;
   LS       CONSTANT NUMBER := 70;
   ST       CONSTANT NUMBER := 255;
   LT       CONSTANT NUMBER := 2000;
   nNewnanCorp       CONSTANT NUMBER := 3000;
   nCommitThreshold  CONSTANT NUMBER := 100;

   nSqlCode NUMBER;
   vSqlErrm VARCHAR2(200);

   -------------------------------------------------------------
   -- Written By  : David Rincon
   -- Tokenizer data types
   -------------------------------------------------------------
   TYPE WordListType IS TABLE OF VARCHAR2(35) INDEX BY BINARY_INTEGER;
   TYPE LineType IS RECORD
    (
         vLine     VARCHAR2(2000),
         WordList  WordListType ,
         bSuccess  BOOLEAN
    );
   TextStructure LineType;
   -------------------------------------------------------------
   -- Written By  : David Rincon
   -------------------------------------------------------------
   PROCEDURE ParseText ( vLine IN VARCHAR2 ) ;
   -------------------------------------------------------------
   -- Written By  : David Rincon
   -------------------------------------------------------------
   PROCEDURE GetWords ( TextStructure IN OUT LineType ) ;
   PRAGMA RESTRICT_REFERENCES ( GETWORDS, WNDS,  WNPS );
   FUNCTION NO_BLANKS ( vLine VARCHAR2 ) RETURN VARCHAR2;
   PRAGMA RESTRICT_REFERENCES ( NO_BLANKS , WNDS,  WNPS );
   -------------------------------------------------------------
   -- Written By  : David Rincon
   -- Function to test if parameter is alphabetic
   -------------------------------------------------------------
   FUNCTION IsAlpha ( cChar CHAR ) RETURN BOOLEAN ;

   -------------------------------------------------------------
   -- Written By  : David Rincon
   -- Function to test if parameter is a digit
   -------------------------------------------------------------
   FUNCTION IsDigit ( cChar CHAR ) RETURN BOOLEAN ;

END plsql_parser;
/
```

```
CREATE OR REPLACE PACKAGE BODY plsql_parser AS
-----------------------------------------------------------------
-- Written By  : David Rincon
--               http://www.cdsiusa.com
--               Copyright Cornerstone Data Systems, INC
-----------------------------------------------------------------
-----------------------------------------------------------------
-- Description : Parser Functions
--
-----------------------------------------------------------------
-- Modification History
--
-- Who          When          What
-- ------------ ------------- ---------------------------------
-- D. Rincon    02/15/15      Created
-----------------------------------------------------------------

   -----------------------------------------------------------------
   -- Written By  : David Rincon
   -- Description : Call the tokenizer function to process
   --               input text lines.
   -----------------------------------------------------------------
   PROCEDURE ParseText ( vLine IN VARCHAR2)
   IS

       TextLine plsql_parser.linetype;

   BEGIN
       TextLine.vLine := vLine;
       GetWords( TextLine );
       FOR i IN 1..TextLine.WordList.COUNT LOOP
           BEGIN
             DBMS_OUTPUT.PUT_LINE( TextLine.WordList(i) ||'<-->');
           END;
       END LOOP;
   END ParseText;
```

```
--------------------------------------------------------------------------
--  FUNCTION NO_BLANKS: ( VARCHAR2 )
--------------------------------------------------------------------------
--  Description   : Receives a string and removes blanks
--
--  Returns       : Return string without blanks
--
--
--------------------------------------------------------------------------
--  Modification History
--  When      Who          What
--  ========  ============ ================
--  02/28/15  David Rincon Created
--------------------------------------------------------------------------
-- Call Syntax : (Pro*C)
-- EXEC SQL EXECUTE
--  Begin
--    x : = NO_BLANKS ( InterfaceStructure );
---         ....
--  End;
-- END-EXEC;
--------------------------------------------------------------------------
 FUNCTION NO_BLANKS ( vLine VARCHAR2 ) RETURN VARCHAR2
 IS
   nLen     NUMBER          := LENGTH( RTRIM(LTRIM(vLine ))) ;
   vStr     VARCHAR2(4000)  := RTRIM(LTRIM(UPPER( vLine )));
   vWord    VARCHAR2(100);
   vRetVal  VARCHAR2(100);
   i        NUMBER;
 BEGIN

   i := 1;
   WHILE ( i <= nLen )  LOOP
       vWord := '';
       WHILE ( SUBSTR(vStr,i,1) NOT IN (' ') ) AND ( i <= nLen ) LOOP
         vWord := vWord || SUBSTR(vStr,i,1);
         i := i + 1;
       END LOOP;
       i := i + 1;
       vRetVal := vRetVal || vWord;
   END LOOP;

   RETURN ( vRetVal ) ;

 END NO_BLANKS ;
```

```
--------------------------------------------------------------------------
--  Procedure GetWords : ( Text Structure  )
--------------------------------------------------------------------------
--  Description   : Receives a string and empty word list, the procedure
--                  will popuplate the list with words derived from string
--
--  Returns       : Modified structure containing word list
--
--
--------------------------------------------------------------------------
--  Modification History
--  When      Who          What
--  ========  ============ ================
--  02/15/15  David Rincon Created
--------------------------------------------------------------------------
-- Call Syntax : (Pro*C)
-- EXEC SQL EXECUTE
--  Begin
--    GetWords ( InterfaceStructure );
---        ....
--  End;
-- END-EXEC;
--------------------------------------------------------------------------
 PROCEDURE GetWords ( TextStructure IN OUT LineType )
 IS
   nLen     NUMBER          := LENGTH( RTRIM(LTRIM(TextStructure.vLine ))) ;
   vStr     VARCHAR2(4000)  := RTRIM(LTRIM(UPPER( TextStructure.vLine )));
   vWord    VARCHAR2(35);
   i        NUMBER;
   j        NUMBER;
 BEGIN

   i := 1;
   j := 1;
   TextStructure.bSuccess := FALSE;
   WHILE ( i <= nLen )  LOOP
       vWord := '';
       WHILE ( SUBSTR(vStr,i,1) NOT IN (' ','/','.') ) AND ( i <= nLen ) LOOP
         vWord := vWord || SUBSTR(vStr,i,1);
         i := i + 1;
       END LOOP;
       IF ( vWord IS NOT NULL ) THEN
            TextStructure.WordList(j) := vWord;
            TextStructure.bSuccess := TRUE;
            j := j + 1;
       END IF;
       i := i + 1;
   END LOOP;

 END GetWords;
```

```
-------------------------------------------------------------
-- Written By  : David Rincon
--               http://www.cdsiusa.com
-------------------------------------------------------------
-------------------------------------------------------------
-- Description : Is character parameter alphabetic ?
--               General purpose function
-------------------------------------------------------------
-- Modification History
--
-- Who          When          What
-- ------------ ------------- ------------------------------
-- D. Rincon    02/15/15      Created
-------------------------------------------------------------
FUNCTION IsAlpha ( cChar CHAR ) RETURN BOOLEAN
IS
 bRetVal BOOLEAN := FALSE;
BEGIN

 IF ( ASCII(cChar) BETWEEN 65 AND 90 ) OR
    ( ASCII(cChar) BETWEEN 97 AND 122 ) THEN
      bRetVal := TRUE;
 END IF;

 RETURN ( bRetVal );

END IsAlpha;
```

```
   ------------------------------------------------------------
   -- Written By  : David Rincon
   --               http://www.cdsiusa.com
   ------------------------------------------------------------
   ------------------------------------------------------------
   -- Description : Is character parameter a digit?
   --               General purpose function
   ------------------------------------------------------------
   -- Modification History
   --
   -- Who          When          What
   -- ------------ ------------- ------------------------------
   -- D. Rincon    02/15/15      Created
   ------------------------------------------------------------
   FUNCTION IsDigit ( cChar CHAR ) RETURN BOOLEAN
   IS
    bRetVal BOOLEAN := FALSE;
   BEGIN

    IF ( ASCII(cChar) BETWEEN 48 AND 57 ) THEN
         bRetVal := TRUE;
    END IF;

    RETURN ( bRetVal );

   END IsDigit;

END plsql_parser;
/
```

USAGE:

```
Compile the above code in schema of your choice.

"plsqlparser.sql" [dos] 265L, 9632C written
cdsi12rac1:oracle:/home/oracle/sql >sqlplus

SQL*Plus: Release 12.1.0.1.0 Production on Fri Oct 9 11:23:13 2015

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Enter user-name: sys as sysdba
Enter password:

Connected to:
Oracle Database 12c Release 12.1.0.1.0 - 64bit Production
With the Real Application Clusters and Automatic Storage Management options

SQL> @plsqlparser

Package created.


Package body created.
SQL> exec plsql_parser.parsetext('Oracle Database 12c introduces a new
multitenant architecture that makes it easy to consolidate many databases
quickly and manage them as a cloud service. Oracle Database 12c also includes
in-memory data processing capabilities delivering breakthrough analytical
performance. Additional database innovations deliver new levels of efficiency,
performance, security, and availability. Oracle Database 12c comes in three
editions to fit your business needs and budget: Enterprise Edition, Standard
Edition, and Standard Edition One.');
>>ORACLE
>>DATABASE
>>12C
>>INTRODUCES
>>A
>>NEW
>>MULTITENANT
>>ARCHITECTURE
>>THAT
>>MAKES
>>IT
>>EASY
>>TO
>>CONSOLIDATE
>>MANY
>>DATABASES
>>QUICKLY
>>AND
>>MANAGE
>>THEM
>>AS
>>A
>>CLOUD
>>SERVICE
>>ORACLE
>>DATABASE
>>12C
>>ALSO
>>INCLUDES
>>IN-MEMORY
>>DATA
>>PROCESSING
>>CAPABILITIES
>>DELIVERING
>>BREAKTHROUGH
>>ANALYTICAL
>>PERFORMANCE
>>ADDITIONAL
>>DATABASE
>>INNOVATIONS
```

```
>>DELIVER
>>NEW
>>LEVELS
>>OF
>>EFFICIENCY,
>>PERFORMANCE,
>>SECURITY,
>>AND
>>AVAILABILITY
>>ORACLE
>>DATABASE
>>12C
>>COMES
>>IN
>>THREE
>>EDITIONS
>>TO
>>FIT
>>YOUR
>>BUSINESS
>>NEEDS
>>AND
>>BUDGET:
>>ENTERPRISE
>>EDITION,
>>STANDARD
>>EDITION,
>>AND
>>STANDARD
>>EDITION
>>ONE

PL/SQL procedure successfully completed.
```