

SQL Dinámico y cursores

Para más información visite <http://www.slinfo.una.ac.cr>

INTRODUCCIÓN

Los procedimientos almacenados en Oracle, permiten invocar fácilmente instrucciones de tipo DML (insert, delete, update) pero son particularmente complejos cuando se invocan consultas DDL (create, alter, drop).

Para poder invocar consultas DDL es necesario utilizar un componente de la base de datos denominado SQL Dinámico, el cual consiste en pasarle mediante cadena de caracteres el comando a ejecutar.

EJEMPLOS DE PROCEDIMIENTOS

Taxonomía de un procedimiento almacenado

Todo procedimiento almacenado en Oracle es precompilado y almacenado en la pool del SGA para su pronta y rápida ejecución. La estructura de un procedimiento almacenado es la siguiente:

```
CREATE [OR REPLACE]  
PROCEDURE <procedure_name> [( <param1> [IN|OUT|IN OUT] <type>,  
                               <param2> [IN|OUT|IN OUT] <type>, ...)]  
IS  
    -- Declaracion de variables locales  
BEGIN  
    -- Sentencias  
[EXCEPTION]  
    -- Sentencias control de excepcion  
END [<procedure_name>;
```

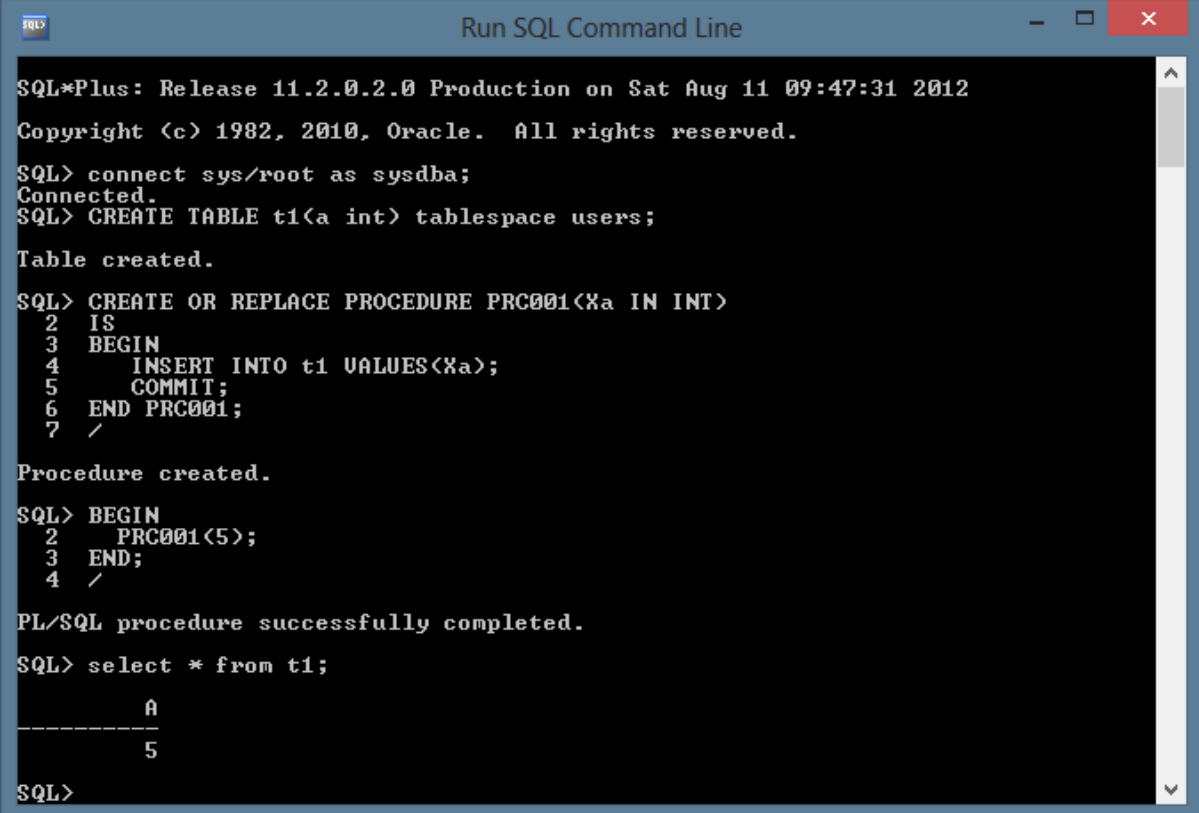
Sin por ejemplo, queremos realizar un procedimiento para que inserte en la tabla t1 es necesario ejecutar el siguiente código:

```
CREATE TABLE T1 (  
    A int  
)tablespace users;
```

```
CREATE OR REPLACE PROCEDURE PRC001(Xa IN INT)
IS
BEGIN
    INSERT INTO t1 VALUES(Xa);
    COMMIT;
END PRC001;
```

Sin queremos ejecutar un procedimiento ya creado previamente debemos encerrarlo entre un bloque begin... end, como se expresa en el siguiente código:

```
BEGIN
    PRC001(5);
END;
```



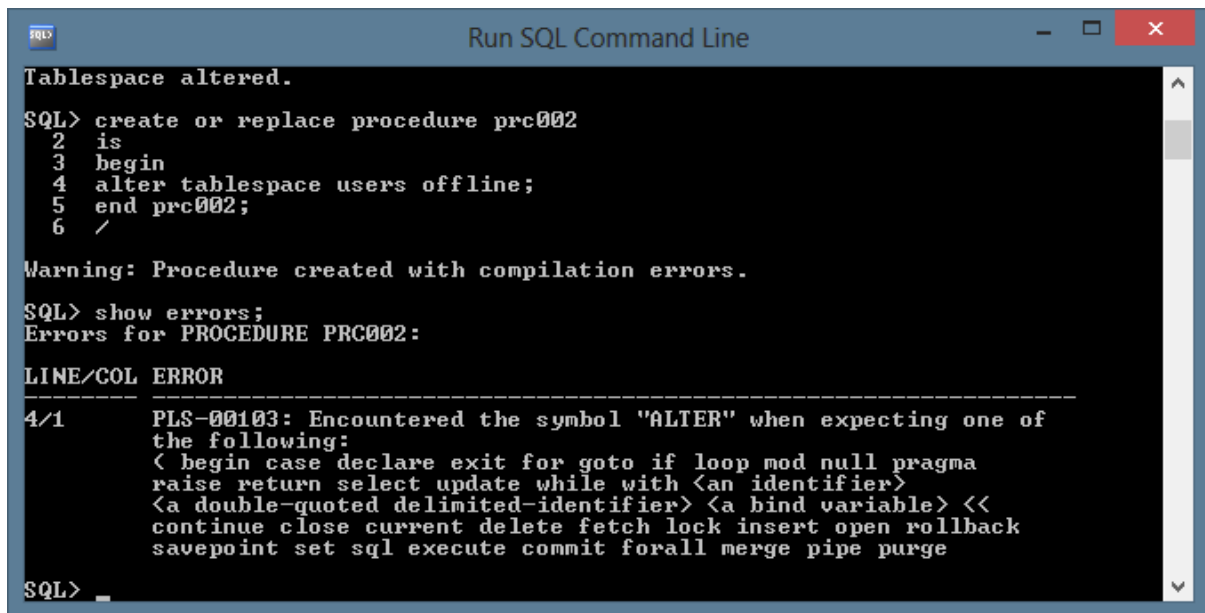
The screenshot shows a window titled "Run SQL Command Line" with a black background and white text. The text displays the following SQL commands and their outputs:

```
SQL*Plus: Release 11.2.0.2.0 Production on Sat Aug 11 09:47:31 2012
Copyright (c) 1982, 2010, Oracle. All rights reserved.
SQL> connect sys/root as sysdba;
Connected.
SQL> CREATE TABLE t1(a int) tablespace users;
Table created.
SQL> CREATE OR REPLACE PROCEDURE PRC001(Xa IN INT)
2  IS
3  BEGIN
4      INSERT INTO t1 VALUES(Xa);
5      COMMIT;
6  END PRC001;
7  /
Procedure created.
SQL> BEGIN
2  PRC001(5);
3  END;
4  /
PL/SQL procedure successfully completed.
SQL> select * from t1;
-----
a
-----
5
SQL>
```

Demostración de DML en procedimientos

Supongamos que se nos ha contratado para crear un procedimiento almacenado capaz de desactivar un tablespace. Sin embargo la misma parece no funcionar.

```
CREATE OR REPLACE PROCEDURE PRC002
IS
BEGIN
    ALTER TABLESPACE users offline;
END PRC002;
```



```
Run SQL Command Line
Tablespace altered.
SQL> create or replace procedure prc002
2  is
3  begin
4  alter tablespace users offline;
5  end prc002;
6  /

Warning: Procedure created with compilation errors.
SQL> show errors;
Errors for PROCEDURE PRC002:
LINE/COL ERROR
-----
4/1      PLS-00103: Encountered the symbol "ALTER" when expecting one of
the following:
< begin case declare exit for goto if loop mod null pragma
raise return select update while with <an identifier>
<a double-quoted delimited-identifier> <a bind variable> <<
continue close current delete fetch lock insert open rollback
savepoint set sql execute commit forall merge pipe purge
SQL>
```

Otro DBA encontró la solución utilizando SQL dinámico mediante la siguiente instrucción:

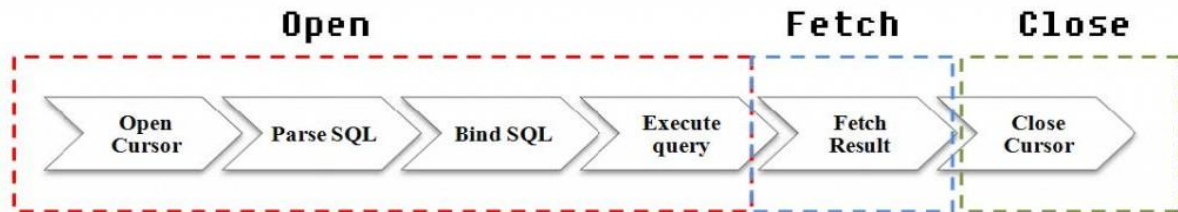
```
CREATE OR REPLACE PROCEDURE PRC002
IS
BEGIN
    EXECUTE IMMEDIATE 'ALTER TABLESPACE users offline ';
END PRC002;
```

Sin embargo por instrucciones de gerencia se ha solicitado crear un procedimiento genérico que tenga como parámetro el nombre del tablespace a desactivar.

```
CREATE OR REPLACE PROCEDURE PRC003 (Xnombre IN varchar2)
IS
    sql_dinamico varchar2(1000);
BEGIN
    sql_dinamico := 'ALTER TABLESPACE :nombre offline';
    EXECUTE IMMEDIATE sql_dinamico USING Xnombre;
END PRC003;
/
```

CURSORES EN ORACLE

El siguiente código es un tipo especial de cursor, simplificado muy semejante al ForEach



```
declare
  cursor cur is
    select col1, col2
    from tabla;
begin
  for v_reg in cur loop
    dbms_output.put_line ('Col1: ' || v_reg.col1 || ' Col2: ' ||
v_reg.col2);
  end loop;
end;
```

Si por ejemplo queremos mostrar todos los datos de una table utilizamos el siguiente procedimiento almacenado:

```
SET SERVEROUTPUT ON;

CREATE OR REPLACE PROCEDURE PRC004
IS
  CURSOR lista IS SELECT a FROM t1;
BEGIN
  FOR actual IN lista LOOP
    dbms_output.put_line('Dato: ' || actual.a);
  END LOOP;
END PRC004;
/
```

Otros recursos:

<http://www.dbanotes.com/database-development/introduction-to-oracle-11g-cursors/>
<http://es.wikibooks.org/wiki/Oracle/PL/SQL/Cursores>