Stored Procedures

What is a Stored Procedure?

A **stored procedure** or in simple a **proc** is a named PL/SQL block which performs one or more specific task. This is similar to a procedure in other programming languages. A procedure has a header and a body. The header consists of the name of the procedure and the parameters or variables passed to the procedure. The body consists or declaration section, execution section and exception section similar to a general PL/SQL Block. A procedure is similar to an anonymous PL/SQL Block but it is named for repeated usage.

We can pass parameters to procedures in three ways.   
1) IN-parameters  
2) OUT-parameters  
3) IN OUT-parameters

A procedure may or may not return any value.

General Syntax to create a procedure is:

*CREATE [OR REPLACE] PROCEDURE proc\_name [list of parameters]*

*IS*

*Declaration section*

*BEGIN*

*Execution section*

*EXCEPTION*

*Exception section*

*END;*

**IS -** marks the beginning of the body of the procedure and is similar to DECLARE in anonymous PL/SQL Blocks. The code between IS and BEGIN forms the Declaration section.

The syntax within the brackets [ ] indicate they are optional. By using CREATE OR REPLACE together the procedure is created if no other procedure with the same name exists or the existing procedure is replaced with the current code.

The below example creates a procedure ‘employer\_details’ which gives the details of the employee.

*1> CREATE OR REPLACE PROCEDURE employer\_details*

*2> IS*

*3> CURSOR emp\_cur IS*

*4> SELECT first\_name, last\_name, salary FROM emp\_tbl;*

*5> emp\_rec emp\_cur%rowtype;*

*6> BEGIN*

*7> FOR emp\_rec in sales\_cur*

*8> LOOP*

*9> dbms\_output.put\_line(emp\_cur.first\_name || ' ' ||emp\_cur.last\_name*

*10> || ' ' ||emp\_cur.salary);*

*11> END LOOP;*

*12>END;*

*13> /*

How to execute a Stored Procedure?

There are two ways to execute a procedure.

1) From the SQL prompt.

*EXECUTE [or EXEC] procedure\_name;*

2) Within another procedure – simply use the procedure name.

*procedure\_name;*

**NOTE:** In the examples given above, we are using backward slash ‘/’ at the end of the program. This indicates the oracle engine that the PL/SQL program has ended and it can begin processing the statements.