**PL/SQL Substitution and Bind Variables**

In addition to PL/SQL variables and constants, you can use variables in two ways. PL/SQL Substitution and Bind Variables allow you pass on values to PL/SQL programs while execution. Lets learn about these two ways

Note: PL/SQL Substitution and Bind variables can only be used for accepting input in SQL\* Plus. Online tool [liveSQL](https://livesql.oracle.com/) do not support PL/SQL Substitution and Bind variables.

PL/SQL Substitution Variables

PL/SQL blocks can accept user input while execution. This provision is available with substitution variables. They are not allocated any memory, so they can’t be used for returning values as output. If you want to return values from a named PL/SQL block then you can use OUT parameters in procedures or return value from functions.

Whenever PL/SQL processor encounters a substitution variable in a block, it halts for user input. All the substitution variables in the block are replaced by the inputted values before sending it to the database.

Substitution variables are used by prefixing a variable name with the ampersand(&) or double ampersand (&&).

**Ampersand(&)**  is used when you want to give a new value each time a substitution variable is encountered in the code. The user has to input value even if the same substitution variable is used at multiple places in a PL/SQL program.

**Double ampersand (&&)** is used when you want the same value to be used for all occurrences of a variable in PL/SQL program. With &&, PL/SQL processor asks only once for each substitution variable. Wherever the substitution variable is encountered in the code it is replaced with previously assigned value.

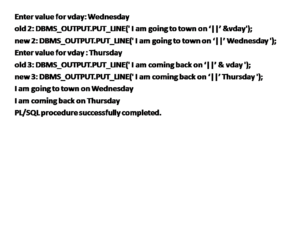
Example with &

BEGIN

DBMS\_OUTPUT.PUT\_LINE(‘I am going to town on ‘||’&vday’);

DBMS\_OUTPUT.PUT\_LINE(‘I am coming back on ‘||‘&vday’);

END;



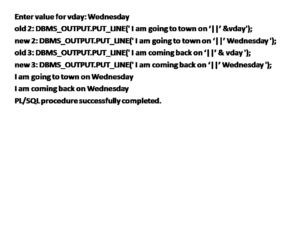
Example with &&

BEGIN

DBMS\_OUTPUT.PUT\_LINE(‘I am going to town on ‘||’&&vday’);

DBMS\_OUTPUT.PUT\_LINE(‘I am coming back on ‘||‘&vday’);

END;



PL/SQL Bind Variables

Bind variable is the facility provided by a host environment like SQL\*Plus. A bind variable can be declared in the host environment and passed as runtime value into PL/SQL code blocks. The bind variable will just be like a variable in a code block which has not been declared in the block using it.

Bind Variables can be created using the command **VARIABLE** just like you declare a variable in PL/SQL code. The Bind variables declared in the host environment can be used in programs prefixed with colon symbol (:)

Example

**Declared in the host environment(SQL\*Plus)**

VARIABLE max\_salary number;

VARIABLE department varchar2(5);

**Used in the PL/SQL program in BEGIN Section**

Select max(salary) into : max\_salary from employees;

So, you have learnt how PL/SQL Substitution and Bind variables can enhance your PL/SQL code blocks. You can make them interactive while execution.

DECLARE

v\_student\_id NUMBER := &sv\_student\_id;

v\_first\_name VARCHAR2(35);

v\_last\_name VARCHAR2(35);

BEGIN

SELECT first\_name, last\_name

INTO v\_first\_name, v\_last\_name

FROM student

WHERE student\_id = v\_student\_id;

DBMS\_OUTPUT.PUT\_LINE ('Student name: '||v\_first\_name||' '||

v\_last\_name);

EXCEPTION

WHEN NO\_DATA\_FOUND

THEN

DBMS\_OUTPUT.PUT\_LINE ('There is no such student');

END;