

Created by Ahsan Arif

## PL/SQL Variable

Here is an example of declaring some variables in an anonymous block:

```
DECLARE
    v_first_name VARCHAR2(20);
    v_last_name  VARCHAR2(20);
    n_employee_id NUMBER;
    d_hire_date DATE;
BEGIN
    NULL;
END;
```

To declare a variable, you type a variable name followed by the data type and terminated by a semicolon (;). You can also explicitly add length constraint to the data type in a set of parentheses.

## PL/SQL Anchors

It refers to the use of keyword %TYPE to declare a variable with the data type is the column data type in a table.

```
DECLARE
    v_first_name  EMPLOYEES.FIRST_NAME%TYPE;
    v_last_name   EMPLOYEES.LAST_NAME%TYPE;
    n_employee_id EMPLOYEES.EMPLOYEE_ID%TYPE;
    d_hire_date    EMPLOYEES.HIRE_DATE%TYPE;
BEGIN
    NULL;
END;
/
```

## Assignment

To assign a value or a variable to a variable in PL/SQL, you use the assignment operator (:=) which is a colon (:) followed by an equal sign (=). See the code listing below to have a better understanding:

```
DECLARE
    v_first_name  EMPLOYEES.FIRST_NAME%TYPE;
    v_last_name   EMPLOYEES.LAST_NAME%TYPE;
    n_employee_id EMPLOYEES.EMPLOYEE_ID%TYPE;
    d_hire_date    EMPLOYEES.HIRE_DATE%TYPE;
```

```

BEGIN
    v_first_name := 'Mary';
    v_last_name  := 'Dose';
    d_hire_date  := TO_DATE('19700101', 'YYYYMMDD');
END;
/

```

You can use INTO of SQL SELECT statement to assign a value to a variable. In this way, the INTO clause move the values from the SELECT query's column list into corresponding PL/SQL variables.

```

SET SERVEROUTPUT ON SIZE 1000000;
DECLARE
    v_first_name EMPLOYEES.FIRST_NAME%TYPE;
    v_last_name  EMPLOYEES.LAST_NAME%TYPE;
    n_employee_id EMPLOYEES.EMPLOYEE_ID%TYPE;
    d_hire_date  EMPLOYEES.HIRE_DATE%TYPE;
BEGIN
    SELECT employee_id,
           first_name,
           last_name,
           hire_date
    INTO n_employee_id,
        v_first_name,
        v_last_name,
        d_hire_date
    FROM employees
    WHERE employee_id = 200;

    DBMS_OUTPUT.PUT_LINE(v_first_name);
    DBMS_OUTPUT.PUT_LINE(v_last_name);
    DBMS_OUTPUT.PUT_LINE(d_hire_date);
END;
/

```

## Initializing Variables

When you declare a variable, its value is uninitialized and hence is NULL. You can initialize variable a value by assigning it a value in declaration section.

```

DECLARE
    n_employee_id EMPLOYEES.EMPLOYEE_ID%TYPE :=200;
    d_hire_date
EMPLOYEES.HIRE_DATE%TYPE:=TO_DATE('19700101', 'YYYYMMDD');
BEGIN
    NULL;

```

END;

/