Kabeer Ahmed
SE-19028

Database Management System (DBMS – 204)

Experiment # 11

Student Name: Kabeer Ahmed	
Roll Number: SE-19028	

Maximum Marks	Performance = 05	Viva = 05	Total = 10
Marks Obtained			
Remarks (if any)			

Experiment evaluated by

Instructor Name:	Engr. Adiba Jafar
Signature and Date:	

Kabeer Ahmed
SE-19028

<u>Outcome</u>
1-Declaring Variables

THEORY

A variable is nothing, but a name given to a storage area that our programs can manipulate. Each variable in PL/SQL has a specific data type, which determines the size and the layout of the variable's memory; the range of values that can be stored within that memory and the set of operations that can be applied to the variable.

The name of a PL/SQL variable consists of a letter optionally followed by more letters, numerals, dollar signs, underscores, and should not exceed 30 characters. By default, variable names are not case-sensitive. You cannot use a reserved PL/SQL keyword as a variable name

PL/SQL Block Structure

`DECLARE` (Optional)
Variables, cursors, user-defined exceptions
`BEGIN` (Mandatory)
– SQL statements
– PL/SQL statements
`EXCEPTION` (Optional)
Actions to perform when errors occur
`END;` (Mandatory)

Executing Statements and PL/SQL Blocks

```
DECLARE
v_variable VARCHAR2(5);
BEGIN
SELECT column_name
INTO v_variable
FROM table_name;
EXCEPTION
WHEN exception_name THEN
...
END;
```

Declaring PL/SQL Variables

PL/SQL variables must be declared in the declaration section or in a package as a global variable. When you declare a variable, PL/SQL allocates memory for the variable's value and the storage location is identified by the variable name.

The syntax for declaring a variable is −

```
variable_name [CONSTANT] datatype [NOT NULL] [:= | DEFAULT initial_value]
```

```
DECLARE
v_hiredate DATE;
v_deptno NUMBER(2) NOT NULL := 10;
v_location VARCHAR2(13) := 'Atlanta';
c_comm CONSTANT NUMBER := 1400;
```

Kabeer Ahmed
SE-19028

Base Scalar Data Types

- **CHAR [(*maximum_length*)]**
- **VARCHAR2 (*maximum_length*)**
- **LONG**
- **LONG RAW**
- **NUMBER [(*precision, scale*)]**
- **BINARY_INTEGER**
- **PLS_INTEGER**
- **BOOLEAN**

```
DECLARE
v_job VARCHAR2(9);
v_count BINARY_INTEGER := 0;
v_total_sal NUMBER(9,2) := 0;
v_orderdate DATE := SYSDATE + 7;
c_tax_rate CONSTANT NUMBER(3,2) := 8.25;
v_valid BOOLEAN NOT NULL := TRUE;
```

Declaring Variables with the %TYPE Attribute

```
identifier Table.column_name%TYPE;
```

```
...
v_name employees.last_name%TYPE;
v_balance NUMBER(7,2);
v_min_balance v_balance%TYPE := 10;
...
```

Kabeer Ahmed
SE-19028

LAB# 11
Declaring Variables

PRACTICE TASKS

1. Evaluate each of the following declarations. Determine which of them are *not* legal and explain why.

A. DECLARE v_id NUMBER(4);

```
DECLARE
        v_id NUMBER(4);
BEGIN
        NULL;
END;
/
```

Ans: **This code is Correct.**

B. DECLARE v_x, v_y, v_z VARCHAR2(10);

```
  DECLARE
V_x , v_y, v_z varchar2(10);
  BEGIN
          NULL;
  END;
  /
```

Ans: **Incorrect, because can't declare multiple variables at once.**

C. DECLARE v_birthdate DATE NOT NULL;

```
  DECLARE
V_BIRTHDATE DATE NOT NULL;
  BEGIN
          NULL;
  END;
  /
```

Ans: **Incorrect, because a NOT NULL variable must be initialized.**

D. DECLARE v_in_stock BOOLEAN := 1;

```
  DECLARE
V_IN_STOCK BOOLEAN:=1;
  BEGIN
          NULL;
  END;
  /
```

Ans: **Incorrect, because a BOOLEAN variable can only have either TRUE or FALSE value.**

Kabeer Ahmed
SE-19028

2. In each of the following assignments, indicate whether the statement is valid and what the valid data type of the result will be.

A. v_days_to_go := v_due_date - SYSDATE;

```
SET SERVEROUTPUT ON
DECLARE
        v_days_to_go NUMBER;
        v_due_date   DATE := SYSDATE + 2;
BEGIN
        v_days_to_go := v_due_date - SYSDATE;
DBMS_OUTPUT.PUT_LINE (v_days_to_go);
END;
/
```

Ans: **Valid, v_days_to_go is of NUMBER type and it's value is 2.**

B. v_sender := USER || ': ' || TO_CHAR(v_dept_no);

```
SET SERVEROUTPUT ON
DECLARE
        v_dept_no NUMBER := 10;
        v_sender  VARCHAR2(60);
BEGIN
        v_sender := USER || ':' ||TO_CHAR(v_dept_no);
        DBMS_OUTPUT.PUT_LINE (v_sender);
END;
/
```

Ans: **Valid, v_sender is of VARCHAR2 type and it's value is 'SCOTT : 10'.**

C. v_flag := TRUE;

```
DECLARE
        v_flag BOOLEAN;
BEGIN
v_flag := True;
END;
/
```

Ans: **Valid, v_flag is of BOOLEAN type and it's value is True.**

D. v_value := NULL;

```
SET SERVEROUTPUT ON
DECLARE
        v_value VARCHAR2(4) := 'TEST';
BEGIN
        v_value := NULL;
        DBMS_OUTPUT.PUT_LINE ('The value is ' || v_value  );
END;
/
```

Ans: **Valid, v_value is of VARCHAR2 type and it's value NULL.**

Kabeer Ahmed
SE-19028

E. v_n1 := v_n2 > (2 * v_n3);

```
        SET SERVEROUTPUT ON
        DECLARE
                    v_n1 BOOLEAN;
                    vn2  NUMBER := 1;
                    v_n3 NUMBER := 2;
        v_result VARCHAR2(5);
   BEGIN
        v_n1 := vn2 > (2 * v_n3);
      IF v_n1 = TRUE THEN v_result := 'True';
      ELSIF v_n1 = FALSE THEN v_result := 'False';
      END IF;
      DBMS_OUTPUT.PUT_LINE (v_result);
   END;
        /
```

Ans: **Valid, v_result is of VARCHAR2 type and it's value is 'False'.**