



## EXERCISE 9: PL/SQL BASICS

Faculty Name: Dr. Balasundaram A  
Slot: L43+L44  
Class Number: CH2021221000708  
Date: 06/10/2021

### DBMS LAB ASSIGNMENT 9 MEHER SHRISHTI NIGAM 20BRS1193

#### PLSQL

1. Write a simple PL/SQL Block to print Hello World.

```
set serverout on;  
BEGIN  
    dbms_output.put_line('Hello World');  
END;  
/
```

```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_9\1.sql  
Hello World  
  
PL/SQL procedure successfully completed.
```

2. Write a simple PL/SQL block to accept two numbers from user and perform all four basic mathematical operations.

```
set serverout on;  
  
DECLARE  
    a integer;  
    b integer;  
    operator varchar2(1);  
    c integer;  
BEGIN  
    a:=&a;  
    b:=&b;  
    dbms_output.put_line('Choose mathematical operation, +, -, *, /');  
    operator:='&operator';  
    IF (operator = '+') THEN  
        c := a + b;  
    ELSIF (operator = '-') THEN  
        c := a - b;  
    ELSIF (operator = '*') THEN  
        c := a * b;  
    ELSIF (operator = '/') THEN  
        c := a / b;  
    END IF;  
    dbms_output.put_line('Ans = '||c);  
END;  
/
```

```

SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_9\2.sql
Enter value for a: 7
old 7:      a:=&a;
new 7:      a:=7;
Enter value for b: 8
old 8:      b:=&b;
new 8:      b:=8;
Enter value for operator: *
old 10:     operator:='&operator';
new 10:     operator:='*';
Choose mathematical operation, +, -, *, /
Ans = 56

PL/SQL procedure successfully completed.

```

3. Write a simple PL/SQL block to fetch the salary of employee named 'Ravi' from the Employee table.

```

set serverout on;
DECLARE
    salary_amt EMP.SALARY%type;
BEGIN
    SELECT SALARY INTO salary_amt FROM EMP WHERE NAME = 'Ravi';
    dbms_output.put_line('Salary of Ravi: '||salary_amt);
END;
/

SQL> CREATE TABLE EMP (ID INT PRIMARY KEY, NAME VARCHAR2(10), DEPT VARCHAR2(10), SALARY INT, AGE INT, SEX VARCHAR2(1));
Table created.

SQL> INSERT ALL
  2 INTO EMP VALUES (1, 'Nelle', 'Grocery', 18900, 28, 'F')
  3 INTO EMP VALUES (2, 'Tom', 'Jewelry', 20100, 37, 'M')
  4 INTO EMP VALUES (3, 'John', 'Stationery', 17900, 17, 'M')
  5 INTO EMP VALUES (4, 'Julian', 'Patio', 19400, 23, 'M')
  6 INTO EMP VALUES (5, 'Lauren', 'Bathroom', 22000, 17, 'F')
  7 INTO EMP VALUES (6, 'Martina', 'Kitchen', 19400, 16, 'F')
  8 INTO EMP VALUES (7, 'Calvin', 'Pharmacy', 21100, 21, 'M')
  9 INTO EMP VALUES (8, 'Ravi', 'Make-up', 19900, 27, 'M')
 10 INTO EMP VALUES (9, 'Josh', 'Bedroom', 19100, 34, 'M')
 11 INTO EMP VALUES (10, 'Tina', 'Apparel', 22900, 30, 'F')
 12 INTO EMP VALUES (11, 'John', 'Grocery', 18000, 38, 'M')
 13 INTO EMP VALUES (12, 'Gavin', 'Jewelry', 21100, 36, 'M')
 14 INTO EMP VALUES (13, 'Hank', 'Kitchen', 18000, 53, 'M')
 15 INTO EMP VALUES (14, 'Rosa', 'Bathroom', 22000, 37, 'F')
 16 SELECT * FROM DUAL;

14 rows created.

SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_9\3.sql
Salary of Ravi: 19900

PL/SQL procedure successfully completed.

```

4. Write a simple PL/SQL block to accept the department number from user and print the count of employees in that department specified by the user.

```

set serverout on;
DECLARE
    dept_id varchar2(2);
    dept_name DEPARTMENT.NAME%type;
    emp_amt integer;
BEGIN
    dept_id := &dept_id;
    SELECT COUNT(NAME) INTO emp_amt FROM EMP WHERE DEPT='3';
    SELECT NAME INTO dept_name FROM DEPARTMENT WHERE DID=dept_id;
    dbms_output.put_line('Number of employees in department '||dept_name||' are '||emp_amt);
END;
/

```

## Declaration of Tables

```
SQL> CREATE TABLE DEPARTMENT (DID VARCHAR(2) PRIMARY KEY, NAME VARCHAR(10), STOCK INT);
```

Table created.

```
SQL> INSERT ALL
```

```
2 INTO DEPARTMENT VALUES ('1', 'Grocery', 25300)
3 INTO DEPARTMENT VALUES ('2', 'Jewelry', 21300)
4 INTO DEPARTMENT VALUES ('3', 'Stationery', 18300)
5 INTO DEPARTMENT VALUES ('4', 'Bathroom', 26200)
6 INTO DEPARTMENT VALUES ('5', 'Pharmacy', 52000)
7 INTO DEPARTMENT VALUES ('6', 'Make-up', 35300)
8 INTO DEPARTMENT VALUES ('7', 'Bedroom', 50300)
9 INTO DEPARTMENT VALUES ('8', 'Apparel', 32300)
10 SELECT * FROM DUAL;
```

8 rows created.

```
SQL>
```

```
SQL>
```

```
SQL> CREATE TABLE EMP (ID INT PRIMARY KEY, NAME VARCHAR2(10), DEPT VARCHAR(2), SALARY INT, AGE INT,
SEX VARCHAR2(1), CONSTRAINT FK_G0 FOREIGN KEY (DEPT) REFERENCES DEPARTMENT(DID));
```

Table created.

```
SQL> INSERT ALL
```

```
2 INTO EMP VALUES (1, 'Nelle', '1', 18900, 28, 'F')
3 INTO EMP VALUES (2, 'Tom', '2', 20100, 37, 'M')
4 INTO EMP VALUES (3, 'John', '3', 17900, 17, 'M')
5 INTO EMP VALUES (4, 'Julian', '3', 19700, 22, 'M')
6 INTO EMP VALUES (5, 'Lauren', '4', 22000, 17, 'F')
7 INTO EMP VALUES (6, 'Martina', '1', 19400, 16, 'F')
8 INTO EMP VALUES (7, 'Calvin', '5', 21100, 21, 'M')
9 INTO EMP VALUES (8, 'Iris', '6', 19900, 27, 'M')
10 INTO EMP VALUES (9, 'Josh', '7', 19100, 34, 'M')
11 INTO EMP VALUES (10, 'Tina', '8', 22900, 30, 'F')
12 INTO EMP VALUES (11, 'John', '1', 18000, 38, 'M')
13 INTO EMP VALUES (12, 'Gavin', '2', 21100, 36, 'M')
14 INTO EMP VALUES (13, 'Hank', '8', 18000, 53, 'M')
15 INTO EMP VALUES (14, 'Rosa', '4', 22000, 37, 'F')
16 SELECT * FROM DUAL;
```

14 rows created.

## Examples of the code running -

```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_9\4.SQL
Enter value for dept_id: 1
old 6:      dept_id := &dept_id;
new 6:      dept_id := 1;
Number of employees in department Grocery are 2

PL/SQL procedure successfully completed.

SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_9\4.SQL
Enter value for dept_id: 5
old 6:      dept_id := &dept_id;
new 6:      dept_id := 5;
Number of employees in department Pharmacy are 2

PL/SQL procedure successfully completed.

SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_9\4.SQL
Enter value for dept_id: 8
old 6:      dept_id := &dept_id;
new 6:      dept_id := 8;
Number of employees in department Apparel are 2

PL/SQL procedure successfully completed.
```

5. Write a simple PL/SQL block to get the age of a person as input and determine if the person is eligible to vote or not.

```
set serverout on;
DECLARE
    age integer;
BEGIN
    age := &age;
    IF(age >= 18) THEN
        dbms_output.put_line('Eligible to vote');
    ELSE
        dbms_output.put_line('Not eligible to vote');
    END IF;
END;
/
```

```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_9\5.sql
Enter value for age: 18
old 4:      age := &age;
new 4:      age := 18;
Eligible to vote

PL/SQL procedure successfully completed.

SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_9\5.SQL
Enter value for age: 6
old 4:      age := &age;
new 4:      age := 6;
Not eligible to vote

PL/SQL procedure successfully completed.

SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_9\5.SQL
Enter value for age: 57
old 4:      age := &age;
new 4:      age := 57;
Eligible to vote

PL/SQL procedure successfully completed.
```

6. Write a simple PL/SQL block to find the greatest of three given numbers.

```
set serverout on;
DECLARE
    a integer;
    b integer;
    c integer;
BEGIN
    a := &a;
    b := &b;
    c := &c;
    IF a > b AND a > c THEN
        dbms_output.put_line('First element, ' || a || ', is the greatest.');
```

```
    ELSIF b > a AND b > c THEN
        dbms_output.put_line('Second element, ' || b || ', is the greatest.');
```

```
    ELSE
        dbms_output.put_line('Third element, ' || c || ', is the greatest.');
```

```
    END IF;
END;
/
```

```

SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_9\6.sql
Enter value for a: 7
old 6:      a := &a;
new 6:      a := 7;
Enter value for b: 10
old 7:      b := &b;
new 7:      b := 10;
Enter value for c: 3
old 8:      c := &c;
new 8:      c := 3;
Second element, 10, is the greatest.

PL/SQL procedure successfully completed.

SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_9\6.sql
Enter value for a: 20
old 6:      a := &a;
new 6:      a := 20;
Enter value for b: 3
old 7:      b := &b;
new 7:      b := 3;
Enter value for c: 7
old 8:      c := &c;
new 8:      c := 7;
First element, 20, is the greatest.

PL/SQL procedure successfully completed.

SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_9\6.sql
Enter value for a: 9
old 6:      a := &a;
new 6:      a := 9;
Enter value for b: 11
old 7:      b := &b;
new 7:      b := 11;
Enter value for c: 45
old 8:      c := &c;
new 8:      c := 45;
Third element, 45, is the greatest.

PL/SQL procedure successfully completed.

```

7. Write a simple PL/SQL block to get the day of the week as input from user and print the corresponding day. 1 – SUN and 7-SAT

```

set serverout on;
DECLARE
    day integer;
BEGIN
    day:=&day;
    CASE day
        when 1 THEN dbms_output.put_line('Monday');
        when 2 THEN dbms_output.put_line('Tuesday');
        when 3 THEN dbms_output.put_line('Wednesday');
        when 4 THEN dbms_output.put_line('Thursday');
        when 5 THEN dbms_output.put_line('Friday');
        when 6 THEN dbms_output.put_line('Saturday');
        when 7 THEN dbms_output.put_line('Sunday');
        ELSE dbms_output.put_line('Invalid');
    END CASE;
END;
/

```

```

SQL>
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_9\7.sql
Enter value for day: 5
old 4:      day:=&day;
new 4:      day:=5;
Friday

PL/SQL procedure successfully completed.

SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_9\7.sql
Enter value for day: 3
old 4:      day:=&day;
new 4:      day:=3;
Wednesday

PL/SQL procedure successfully completed.

SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_9\7.sql
Enter value for day: 9
old 4:      day:=&day;
new 4:      day:=9;
Invalid

PL/SQL procedure successfully completed.

```

9. Write a simple PL/SQL block to print the sum of first n natural numbers using for loop.

```

set serverout on;
DECLARE
    n integer;
    s integer;
BEGIN
    n:=&n;
    s := 0;
    FOR i in 1..n
    LOOP
        s := s + i;
    END LOOP;
    dbms_output.put_line('Sum of first ' || n || ' natural numbers: ' || s);
END;
/

```

```

PL/SQL procedure successfully completed.

SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_9\8.sql
Enter value for n: 5
old 5:      n:=&n;
new 5:      n:=5;
Sum of first 5 natural numbers: 15

PL/SQL procedure successfully completed.

SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_9\8.sql
Enter value for n: 1000
old 5:      n:=&n;
new 5:      n:=1000;
Sum of first 1000 natural numbers: 500500

PL/SQL procedure successfully completed.

SQL> 

```

10. Write a simple PL/SQL block to print the sum of first n natural numbers using while loop.

```
set serverout on;
DECLARE
    n integer;
    i integer;
    s integer;
BEGIN
    n:=&n;
    s := 0;
    i := 1;
    WHILE i <= n
    LOOP
        s := s + i;
        i := i + 1;
    END LOOP;
    dbms_output.put_line('Sum of first ' || n || ' natural numbers: ' || s);
END;
/
```

```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_9\9.sql
Enter value for n: 5
old 6:      n:=&n;
new 6:      n:=5;
Sum of first 5 natural numbers: 15

PL/SQL procedure successfully completed.

SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_9\9.sql
Enter value for n: 1000
old 6:      n:=&n;
new 6:      n:=1000;
Sum of first 1000 natural numbers: 500500

PL/SQL procedure successfully completed.
```