

EXERCISE 10: PL/SQL PROCEDURES AND FUNCTIONS

Faculty Name: Dr. Balasundaram A
Slot: L43+L44
Class Number: CH2021221000708
Date: 13/10/2021

DBMS LAB ASSIGNMENT 10 MEHER SHRISHTI NIGAM 20BRS1193

PLSQL PROCEDURES AND FUNCTIONS

1. Write a PL/SQL function to print Hello world.

Function Code:

```
CREATE OR REPLACE FUNCTION HELLO  
RETURN VARCHAR2  
IS  
P VARCHAR2(30);  
BEGIN  
    P:='HELLO WORLD';  
RETURN P;  
END HELLO;  
/
```

Main Code:

```
set serverout on;  
DECLARE  
    V_OUT VARCHAR2(30);  
BEGIN  
    V_OUT := HELLO();  
    dbms_output.put_line('output string: ' || v_out);  
END;  
/
```

```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\1.sql  
  
Function created.  
  
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\2.sql  
output string: HELLO WORLD  
  
PL/SQL procedure successfully completed.
```

2. Write a PL/SQL function that returns the total number of employees present.

Function Code:

```
CREATE OR REPLACE FUNCTION COUNT_EMP
RETURN INTEGER
IS
TOTAL INTEGER;
BEGIN
    SELECT COUNT(*) INTO TOTAL FROM EMP;
RETURN TOTAL;
END COUNT_EMP;
/
```

Main Code:

```
set serverout on;
DECLARE
    V_OUT INTEGER;
BEGIN
    V_OUT := COUNT_EMP();
    dbms_output.put_line('Total Number of Employees: ' || v_out);
END;
/
```

EMP table:

```
SQL> CREATE TABLE EMP (ID INT PRIMARY KEY, NAME VARCHAR2(10), DEPT VARCHAR(2), SALARY INT, AGE INT, SEX VARCHAR2(1));
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.
```

```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\2_1.sql
```

```
Function created.
```

```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\2_2.sql
```

```
Total Number of Employees: 14
```

```
PL/SQL procedure successfully completed.
```

3. Write a PL/SQL function that displays the course description for a course. If the course is not available, suitable message should be displayed stating that the course is not available.

Function Code:

```
CREATE OR REPLACE FUNCTION COURSE_DESC
(COURSE_NO COURSES.ID%TYPE)
RETURN VARCHAR2
IS
    C_DESCRIPTION VARCHAR2(200);
BEGIN
    SELECT DESCRIPTION INTO C_DESCRIPTION FROM COURSES WHERE ID = COURSE_NO;
RETURN C_DESCRIPTION;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN('The Course is not in the database');
END COURSE_DESC;
/
```

Main Code:

```
set serverout on;
DECLARE
    C_OUT COURSES.DESCRPTION%TYPE;
    COURSE_NO COURSES.ID%TYPE;
BEGIN
    COURSE_NO := &COURSE_NO;
    C_OUT := COURSE_DESC(COURSE_NO);
    dbms_output.put_line('Description of course: ' || COURSE_NO || ' is: ' || C_OUT);
END;
/
```

Courses Table:

```
SQL> CREATE TABLE COURSES (ID INT PRIMARY KEY, CODE VARCHAR2(10), NAME VARCHAR2(50), DESCRIPTION VARCHAR(200), CREDITS INT);
Table created.

SQL> INSERT ALL
  2 INTO COURSES VALUES (1, 'CSE1001', 'Programming Basics', 'Programming is the process of creating a set of instructions that tell a computer how to perform a task.', 4)
  3 INTO COURSES VALUES (2, 'CSE1002', 'Object Oriented Programming', 'Object-oriented programming (OOP) is a computer programming model that organizes software design around data, or objects, rather than functions and logic.', 4)
  4 INTO COURSES VALUES (3, 'CSE1003', 'Networks and Communication', 'Network communication, or internetworking, defines a set of protocols that allow application programs to talk with each other.', 4)
  5 INTO COURSES VALUES (4, 'CSE2001', 'Digital Logic And Design', 'Digital electronic circuits operate with voltages of two logic levelsnamely Logic Low and Logic High.', 3)
  6 INTO COURSES VALUES (5, 'CSE2002', 'Data Structures and Algorithms', 'A data structure is a named location that can be used to store and organize data. And, an algorithm is a collection of steps to solve a particular problem.', 3)
  7 INTO COURSES VALUES (6, 'CSE2003', 'Operating Systems', 'An operating system (OS) is system software that manages computer hardware, software resources, and provides common services for computer programs.', 4)
  8 INTO COURSES VALUES (7, 'ECE1001', 'Analog Electronics', 'Analogue electronics are electronic systems with a continuously variable signal.', 4)
  9 INTO COURSES VALUES (8, 'ECE1002', 'Signal and System', 'Signals and Systems is an introduction to analog and digital signal processing, a topic that forms an integral part of engineering systems in many diverse areas.', 3)
 10 INTO COURSES VALUES (9, 'ECE1003', 'Electromagnetic Theory', 'Electromagnetic theory based on Maxwell's equations establishes the basic principle of electrical and electronic circuits over the entire frequency spectrum from dc to optics.', 4)
 11 INTO COURSES VALUES (10, 'ECE2001', 'Sensors and Actuators', 'A sensor is a device that changes a physical parameter to an electrical output. As against, an actuator is a device that converts an electrical signal to a physical output.', 4)
 12 SELECT * FROM DUAL;
```

10 rows created.

```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\3_1.sql

Function created.
```

```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\3_2.sql
Enter value for course_no: 9
old 5: COURSE_NO := &COURSE_NO;
new 5: COURSE_NO := 9;
Description of course: 9 is: Electromagnetic theory based on Maxwells equations
establishes the basic principle of electrical and electronic circuits over the
entire frequency spectrum from dc to optics.

PL/SQL procedure successfully completed.

SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\3_2.sql
Enter value for course_no: 11
old 5: COURSE_NO := &COURSE_NO;
new 5: COURSE_NO := 11;
Description of course: 11 is: The Course is not in the database

PL/SQL procedure successfully completed.

SQL> _
```

4. Write a PL/SQL procedure to print Hello world.

Procedure Code:

```
set serverout on;
CREATE OR REPLACE PROCEDURE P_HELLOWORLD
IS
BEGIN
    dbms_output.put_line('Hello World!!');
END;
/
```

Direct Execution:

```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\4_1.sql

Procedure created.

SQL> exec P_HELLOWORLD;
Hello World!!

PL/SQL procedure successfully completed.
```

Main Code for explicit procedure call from block using a sql file:

```
BEGIN
P_HELLOWORLD();
END;
/
```

```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\4_2.sql
Hello World!!

PL/SQL procedure successfully completed.
```

5. Write a PL/SQL procedure to accept name as input parameter and print a greeting message.

Procedure Code:

```
CREATE OR REPLACE PROCEDURE GREETING(NAME IN VARCHAR2)
IS
BEGIN
    dbms_output.put_line('Hello ' || NAME || ' !! Welcome to PLSQL.');
```

Direct Execution:

```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\5_1.sql

Procedure created.

SQL> exec GREETING('Neeraj Chopra');
Hello Neeraj Chopra !! Welcome to PLSQL.

PL/SQL procedure successfully completed.
```

Main Code for explicit procedure call from block using a sql file:

```
set serverout on;
DECLARE
    MY_NAME VARCHAR2(100);
BEGIN
    MY_NAME := '&MY_NAME';
    GREETING(MY_NAME);
END;
/
```

```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\5_2.sql
Enter value for my_name: Meher Shrishti Nigam
old 4:      MY_NAME := '&MY_NAME';
new 4:      MY_NAME := 'Meher Shrishti Nigam';
Hello Meher Shrishti Nigam !! Welcome to PLSQL.

PL/SQL procedure successfully completed.

SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\5_2.sql
Enter value for my_name: Nancy Drew
old 4:      MY_NAME := '&MY_NAME';
new 4:      MY_NAME := 'Nancy Drew';
Hello Nancy Drew !! Welcome to PLSQL.

PL/SQL procedure successfully completed.
```

6. Write a PL/SQL procedure that sets a greeting message to the output parameter. Invoke the procedure and observe the output.

Procedure Code:

```
CREATE OR REPLACE PROCEDURE GREETING_OUT(GREETING_VAR OUT VARCHAR2)
IS
BEGIN
    GREETING_VAR := 'Hello User! Welcome to PLSQL.';
END;
/
```

Main Code:

```
DECLARE
    GREET_VAR VARCHAR2(100);
BEGIN
    GREETING_OUT(GREET_VAR);
    dbms_output.put_line(GREET_VAR);
END;
/
```

```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\6_1.sql
Procedure created.

SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\6_2.sql
Hello User! Welcome to PLSQL.

PL/SQL procedure successfully completed.
```

7. Write a PL/SQL procedure using IN OUT parameter to display a greeting message.

Procedure Code:

```
CREATE OR REPLACE PROCEDURE GREETING_IN_OUT(GREETING_VAR IN OUT VARCHAR2)
IS
BEGIN
    GREETING_VAR := 'Hello ' || GREETING_VAR || '! Welcome to PLSQL.';
END;
/
```

Main Code:

```
DECLARE
    GREET_VAR VARCHAR2(100);
BEGIN
    GREET_VAR := '&GREET_VAR';
    GREETING_IN_OUT(GREET_VAR);
    dbms_output.put_line(GREET_VAR);
END;
/
```

```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\7_1.sql
Procedure created.

SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\7_2.sql
Enter value for greet_var: Meher Shrishti Nigam
old 4:      GREET_VAR := '&GREET_VAR';
new 4:      GREET_VAR := 'Meher Shrishti Nigam';
Hello Meher Shrishti Nigam! Welcome to PLSQL.

PL/SQL procedure successfully completed.

SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\7_2.sql
Enter value for greet_var: Rohit Sharma
old 4:      GREET_VAR := '&GREET_VAR';
new 4:      GREET_VAR := 'Rohit Sharma';
Hello Rohit Sharma! Welcome to PLSQL.

PL/SQL procedure successfully completed.
```

8. Write a PL/SQL procedure to compute the employee bonus and print the same.

Procedure Code:

```
CREATE OR REPLACE PROCEDURE CALC_BONUS
(EMP_ID IN NUMBER, BONUS OUT NUMBER)
IS
BEGIN
    SELECT SALARY*0.10 INTO BONUS FROM EMP WHERE ID = EMP_ID;
END CALC_BONUS;
/
```

Main Code:

```
set serverout on;
DECLARE
    BONUS INTEGER;
    E_ID INTEGER;
BEGIN
    E_ID := &EMPLOYEE_NO;
    CALC_BONUS(E_ID, BONUS);
    dbms_output.put_line('The Bonus for the employee' || E_ID || ' is ' || BONUS);
END;
/
```

EMP Table:

```
SQL> CREATE TABLE EMP (ID INT PRIMARY KEY, NAME VARCHAR2(10), DEPT VARCHAR(2), SALARY INT, AGE INT, SEX VARCHAR2(1));
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.
```

```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\8_1.sql

Procedure created.

SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\8_2.sql
Enter value for employee_no: 1
old   5:      E_ID := &EMPLOYEE_NO;
new   5:      E_ID := 1;
The Bonus for the employee1 is 1890

PL/SQL procedure successfully completed.

SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\8_2.sql
Enter value for employee_no: 8
old   5:      E_ID := &EMPLOYEE_NO;
new   5:      E_ID := 8;
The Bonus for the employee8 is 1990

PL/SQL procedure successfully completed.

SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\8_2.sql
Enter value for employee_no: 12
old   5:      E_ID := &EMPLOYEE_NO;
new   5:      E_ID := 12;
The Bonus for the employee12 is 2110

PL/SQL procedure successfully completed.
```

9. Write a PL/SQL procedure for inserting values into student table.

Procedure Code:

```
CREATE OR REPLACE PROCEDURE AddStudent
(i_id student.id%type,
i_first_name student.first_name%type,
i_last_name student.last_name%type,
i_major student.major%type)
IS
BEGIN
    INSERT INTO student (id, first_name, last_name, major, credits) values (i_id,
i_first_name, i_last_name, i_major, 0);
END AddStudent;
/
```

Main Code:

```
set serverout on;
DECLARE
    s_id student.id%type;
    s_first_name student.first_name%type;
    s_last_name student.last_name%type;
    s_major student.major%type;
BEGIN
    s_id := &s_id;
    s_first_name := '&s_first_name';
    s_last_name := '&s_last_name';
    s_major := '&s_major';
    AddStudent(s_id, s_first_name, s_last_name, s_major);
END;
/
```

```
SQL> CREATE TABLE STUDENT (ID INT PRIMARY KEY, first_name VARCHAR2(10), last_name VARCHAR2(10), major VARCHAR(10), credits INT);
Table created.
```

```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\9_1.sql
Procedure created.

SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\9_2.sql
Enter value for s_id: 1
old 7:      s_id := &s_id;
new 7:      s_id := 1;
Enter value for s_first_name: Meher
old 8:      s_first_name := '&s_first_name';
new 8:      s_first_name := 'Meher ';
Enter value for s_last_name: Nigam
old 9:      s_last_name := '&s_last_name';
new 9:      s_last_name := 'Nigam';
Enter value for s_major: BRS
old 10:     s_major := '&s_major';
new 10:     s_major := 'BRS';

PL/SQL procedure successfully completed.
```



```

SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\9_2.sql
Enter value for s_id: 2
old 7:      s_id := &s_id;
new 7:      s_id := 2;
Enter value for s_first_name: Rohit
old 8:      s_first_name := '&s_first_name';
new 8:      s_first_name := 'Rohit';
Enter value for s_last_name: Sharma
old 9:      s_last_name := '&s_last_name';
new 9:      s_last_name := 'Sharma';
Enter value for s_major: ECE
old 10:     s_major := '&s_major';
new 10:     s_major := 'ECE';

PL/SQL procedure successfully completed.

SQL> SELECT * FROM STUDENT;

      ID FIRST_NAME LAST_NAME MAJOR CREDITS
-----
      1 Meher      Nigam    BRS      0
      2 Rohit      Sharma   ECE      0

SQL>

```

10. Write a PL/SQL procedure to check if an employee exists in database and throw suitable exception message if not present.

Procedure Code:

```

CREATE OR REPLACE PROCEDURE EMPLOYEE_EXIST(NAME_IN IN VARCHAR2, CODE OUT VARCHAR2)
IS
    E_ID INTEGER;
BEGIN
    SELECT ID INTO E_ID FROM EMP WHERE NAME = NAME_IN;
    CODE := NAME_IN || ' was found in the database.';
EXCEPTION
WHEN NO_DATA_FOUND THEN
    CODE := NAME_IN || ' is not present in the database.';
END EMPLOYEE_EXIST;
/

```

Main Code:

```

set serverout on;
DECLARE
    E_NAME VARCHAR2(100);
    E_CODE VARCHAR2(100);
BEGIN
    E_NAME := '&E_NAME';
    EMPLOYEE_EXIST(E_NAME, E_CODE);
    dbms_output.put_line(E_CODE);
END;
/

```

```

SQL> CREATE TABLE EMP (ID INT PRIMARY KEY, NAME VARCHAR2(10), DEPT VARCHAR(2), SALARY INT, AGE INT, SEX VARCHAR2(1));
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.

```

```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\10_1.sql
```

Procedure created.

```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\10_2.sql
```

```

Enter value for e_name: Iris
old   5:      E_NAME := '&E_NAME';
new   5:      E_NAME := 'Iris';
Iris was found in the database.

```

PL/SQL procedure successfully completed.

```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_10\10_2.sql
```

```

Enter value for e_name: Meher
old   5:      E_NAME := '&E_NAME';
new   5:      E_NAME := 'Meher';
Meher is not present in the database.

```

PL/SQL procedure successfully completed.