EXERCISE 6: PL/SQL PROCEDURES AND FUNCTIONS

Name : Mahesh Jagtap
Reg No. 24MCS1017
Date: 23.08.2024

Answer all questions.

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

```
SQL> CREATE OR REPLACE FUNCTION print_hello_world
  2  RETURN VARCHAR2
  3  IS
  4  BEGIN
  5     RETURN 'Hello World';
  6  END;
  7  /

Function created.
```

```
SQL> DECLARE
  2     v_result VARCHAR2(50);
  3  BEGIN
  4     v_result := print_hello_world;
  5     DBMS_OUTPUT.PUT_LINE(v_result);
  6  END;
  7  /
Hello World

PL/SQL procedure successfully completed.
```

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

```
SQL> CREATE TABLE employees_24mcs1017 (
  2       employee_id NUMBER PRIMARY KEY,
  3       name VARCHAR2(50),
  4       department_id NUMBER,
  5       salary NUMBER
  6  );

Table created.
```

```
SQL> CREATE OR REPLACE FUNCTION total_employees
  2  RETURN NUMBER
  3  IS
  4     v_count NUMBER;
  5  BEGIN
  6     SELECT COUNT(*) INTO v_count FROM employees_24mcs1017;
  7     RETURN v_count;
  8  END;
  9  /

Function created.
```

```
SQL> DECLARE
  2     v_total NUMBER;
  3  BEGIN
  4     v_total := total_employees;
  5     DBMS_OUTPUT.PUT_LINE('Total employees: ' || v_total);
  6  END;
  7  /
Total employees: 4

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, a suitable message should be displayed stating that the course is not available.

```
CREATE OR REPLACE FUNCTION get_course_description(p_course_id IN VARCHAR2)
RETURN VARCHAR2
IS
  v_description VARCHAR2(200);
BEGIN
  SELECT description INTO v_description
  FROM courses_24mcs1017
  WHERE course_id = p_course_id;

  RETURN v_description;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
        RETURN 'Course not available';
END;
/
```

```
SQL> DECLARE
  2     v_description VARCHAR2(200);
  3  BEGIN
  4     v_description := get_course_description('C101');
  5     DBMS_OUTPUT.PUT_LINE('Course Description: ' || v_description);
  6  END;
  7  /
Course Description: software engineering principles, and the role of computers
in society.

PL/SQL procedure successfully completed.
```

```
SQL> DECLARE
  2     v_description VARCHAR2(200);
  3  BEGIN
  4     v_description := get_course_description('C105');
  5     DBMS_OUTPUT.PUT_LINE('Course Description: ' || v_description);
  6  END;
  7  /
Course Description: Course not available

PL/SQL procedure successfully completed.
```

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

```
SQL> CREATE OR REPLACE PROCEDURE proc_print_hello_world
  2  IS
  3  BEGIN
  4     DBMS_OUTPUT.PUT_LINE('Hello World');
  5  END;
  6  /

Procedure created.

SQL> BEGIN
  2     proc_print_hello_world;
  3  END;
  4  /
Hello World

PL/SQL procedure successfully completed.
```

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

```
SQL> CREATE OR REPLACE PROCEDURE greet_user(p_name IN VARCHAR2)
  2  IS
  3  BEGIN
  4     DBMS_OUTPUT.PUT_LINE('Hello, ' || p_name || '.Good Morning');
  5  END;
  6  /

Procedure created.

SQL> BEGIN
  2     greet_user('Mahesh');
  3  END;
  4  /
Hello, Mahesh.Good Morning

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.

```
SQL> CREATE OR REPLACE PROCEDURE set_greeting(p_name IN VARCHAR2, p_greeting OUT VARCHAR2)
  2  IS
  3  BEGIN
  4     p_greeting := 'Hello, ' || p_name || ' Good Evening !';
  5  END;
  6  /

Procedure created.

SQL> DECLARE
  2     v_greeting VARCHAR2(100);
  3  BEGIN
  4     set_greeting('Abhi', v_greeting);
  5     DBMS_OUTPUT.PUT_LINE(v_greeting);
  6  END;
  7  /
Hello, Abhi Good Evening !

PL/SQL procedure successfully completed.
```

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

```
SQL> CREATE OR REPLACE PROCEDURE display_greeting(p_name IN OUT VARCHAR2)
  2  IS
  3  BEGIN
  4     p_name := 'Hello, ' || p_name || ' Nice to meet you !';
  5     DBMS_OUTPUT.PUT_LINE(p_name);
  6  END;
  7  /

Procedure created.

SQL> DECLARE
  2     v_name VARCHAR2(100) := 'Mahesh';
  3  BEGIN
  4     display_greeting(v_name);
  5  END;
  6  /
Hello, Mahesh Nice to meet you !

PL/SQL procedure successfully completed.
```

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

```
SQL> CREATE OR REPLACE PROCEDURE compute_employee_bonus (p_employee_id IN NUMBER)
  2  IS
  3     v_salary NUMBER(10, 2);
  4     v_bonus  NUMBER(10, 2);
  5  BEGIN
  6     SELECT salary INTO v_salary
  7     FROM employee_24mcs1017
  8     WHERE employee_id = p_employee_id;
  9
 10     v_bonus := v_salary * 0.10;
 11
 12     DBMS_OUTPUT.PUT_LINE('Employee ID: ' || p_employee_id || ', Bonus: ' || v_bonus);
 13  EXCEPTION
 14     WHEN NO_DATA_FOUND THEN
 15        DBMS_OUTPUT.PUT_LINE('Employee ID ' || p_employee_id || ' not found.');
 16  END;
 17  /

Procedure created.
```

```
SQL> BEGIN
  2     compute_employee_bonus(101);
  3  END;
  4  /
Employee ID: 101, Bonus: 5040

PL/SQL procedure successfully completed.
```

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

```
SQL> CREATE TABLE student_24mcs1017(
  2      student_id NUMBER PRIMARY KEY,
  3      first_name VARCHAR2(50),
  4      last_name VARCHAR2(50),
  5      course_id VARCHAR2(10)
  6  );

Table created.
```

```
SQL> CREATE OR REPLACE PROCEDURE insert_student (
  2     p_student_id IN NUMBER,
  3     p_first_name IN VARCHAR2,
  4     p_last_name IN VARCHAR2,
  5     p_course_id IN VARCHAR2
  6  )
  7  IS
  8  BEGIN
  9     INSERT INTO student_24mcs1017 (student_id, first_name, last_name, course_id)
 10     VALUES (p_student_id, p_first_name, p_last_name, p_course_id);
 11
 12     DBMS_OUTPUT.PUT_LINE('Student inserted successfully.');
 13  EXCEPTION
 14     WHEN DUP_VAL_ON_INDEX THEN
 15        DBMS_OUTPUT.PUT_LINE('Error: Student ID already exists.');
 16     WHEN OTHERS THEN
 17        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
 18  END;
 19  /

Procedure created.

SQL> BEGIN
  2     insert_student(1001, 'Virat', 'Kohli', 'C101');
  3  END;
  4  /
Student inserted successfully.

PL/SQL procedure successfully completed.
```

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

```
SQL> CREATE OR REPLACE PROCEDURE check_employee_exists(p_employee_id IN NUMBER)
  2  IS
  3     v_count NUMBER;
  4     employee_not_found EXCEPTION;
  5  BEGIN
  6     SELECT COUNT(*)
  7     INTO v_count
  8     FROM employee_24mcs1017
  9     WHERE employee_id = p_employee_id;
 10
 11     IF v_count = 0 THEN
 12        RAISE employee_not_found;
 13     ELSE
 14        DBMS_OUTPUT.PUT_LINE('Employee ID ' || p_employee_id || ' exists in the database.');
 15     END IF;
 16
 17  EXCEPTION
 18     WHEN employee_not_found THEN
 19        DBMS_OUTPUT.PUT_LINE('Error: Employee ID ' || p_employee_id || ' does not exist.');
 20     WHEN OTHERS THEN
 21        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
 22  END;
 23  /

Procedure created.
```

```
SQL> BEGIN
  2      check_employee_exists(101);
  3      check_employee_exists(999);
  4  END;
  5  /
Employee ID 101 exists in the database.
Error: Employee ID 999 does not exist.

PL/SQL procedure successfully completed.
```