PROCEDURES/FUNCTIONS

Q.1 Explain the usage of IN OUT parameter of a PL/SQL procedure with the help of an example.

Ans. An IN OUT parameter pass initial values to the procedure being called and return updated values to the caller subprogram. Inside the procedure, an IN OUT parameter acts like an initialized variable. So the actual parameter must be a variable, it can not be a constant or an expression.

For example, if we have to write a procedure, which receives salary as parameter and increases it by 10% if it less then 5000 otherwise increases it by 5%.

```
CREATE OR REPLACE PROCEDURE salary_increment (salary IN OUT NUMBER) AS
BEGIN
        IF salary > 5000 THEN
                Salary : = Salary +Salary*0.10;
        ELSE
                Salary : = Salary+ Salary *0.05;
        END IF;
END;
```

Q.2 Why are named procedure referred to as stored procedures?

Ans: The named procedures are referred to as stored procedures because the named procedures are compiled and stored as schema objects in the Oracle database. In contrast the local or anonymous procedures are compiled at the time of their execution and are not saved as part of database.

Q.3 What are actual parameters? What are formal parameters?

Ans. The variable or literals listed in the procedure call statement are called actual parameter, whereas formal parameters are the ones that are listed in the procedure header and are used in procedure definition.

Q.4 What are the advantages of stored procedures?

Ans. The advantages are –

(i) They are stored in compiled form and thus save on execution time.

(ii) They can accept parameter therefore they are flexible in nature.

(iii) They are stored in database and hence are accessible to various applications that can connect to Oracle.

Q.5 What is following procedure doing:

```
        CREATE OR REPLACE FUNCTION getBDate ( v_ssn VARCHAR2)
        RETURN DATE
        AS
                v_bdate  employee.bdate%TYPE;
        BEGIN
                SELECT bdate INTO v_bdate FROM employee
                WHERE ssn = v_ssn;
                RETURN v_bdate:
        END;
```

Ans. The function gets the serial number ssn of an employee as an IN parameter, reads the corresponding values for the field bdate and return the read value.

Q.6 Find the errors in the following fragment code:

```
        CREATE OR REPLACE PROCEDURE Test1
        AS
        DECLARE
                A NUMBER (2);
                B NUMBER (2);
                C NUMBER (4);
        BEGIN
                A: = &A;
                B: = &B;
                C =A+B;
                RETURN C;
        END;
```

Ans. The errors are –

(i) Use of keyword DECLARE is illegal, local declarations at procedure level are made without using the DECLARE clause.

(ii) The operator : = should be used instead of = for assigning value to variable C. Thus the corrected statement will be- C : = A+B

(iii) RETURN statement of a procedure can not return a value.

Q.7 What are functions? How are they different from procedures?
Ans. Functions are the modules that carry out one specific job and return a value.
Difference:
A Function like a procedure received arguments from the calling program. The difference is that a function is part of an expression and return a single value to the calling program for its use whereas a procedure carries out some specific job but does not return any value.

Q. 8 Name the different types of modes formal parameters can have in a PL/SQL procedure. If the mode is not specified in a procedure what will be the default mode?

OR

Q. Differentiate between the IN and OUT parameters of a PL/SQL procedure.
Ans : Different ways in which parameters can be passed to stored procedure are:
IN : It is the default way of passing parameter to a procedure, where the formal parameter acts like a constant and the actual parameter can be a constant, variables, expression or literal.

OUT: It is used to return values to caller, the formal parameter acts like an un – initialized variable and the actual parameter must be a variable.

IN OUT: It is used to pass as well as return values, the formal parameter acts like an initialized variable and the actual parameter must be a variable.

If the mode is not specified in a procedure, then IN is assumed to be the mode of the parameter (default mode).

Q. 9 Write a PL/SQL procedure to return a value for finding the sum of first 10 natural number using OUT parameter.
Ans

```
        CREATE OR REPLACE PROCEDURE SUM_NATURAL (sum  OUT  NUMBER) AS
        BEGIN
                sum  : =  0;
                FOR  I  IN  1. .10
                LOOP
                        sum : = sum + I;
                END LOOP;
        END;
```

Q. 10 Write a PL/SQL procedure called NEXTMONTH that takes a date as parameter and adds 30 days to that date and displays it.

```
Ans     CREATE OR REPLACE PROCEDURE NEXTMONTH (Dt  DATE)  AS
        Ndt    DATE;
        BEGIN
```

```
    SELECT Dt + 30 INTO Ndt FROM dual;
    DBMS_OUTPUT.PUT_LINE( ‘The Date after 30 days will be’ || ndt);
END;
```

Q.11 Write a PL/SQL procedure called FACTORIAL that takes an integer as parameter, find its factorial and display it. (For example, factorial of 3 = 3*2*1 =6).

Ans:

```
CREATE OR REPLACE PROCEDURE FACTORIAL (NUM NUMBER) AS
FACT  NUMBER : = 1;
BEGIN
    FOR  I  IN  1. .NUM
    LOOP
        FACT : = FACT * I;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE (‘FACTORIAL OF’ || NUM||  ‘IS’ || FACT);
END;
```

Q.12 Write a PL/SQL procedure called MULTI_TABLE that takes two numbers are parameter and displays the multiplication of the first parameter till the second parameter.

Ans.

```
CREATE OR REPLACE PROCEDURE MULTI_TABLE (a  NUMBER, b  NUMBER) AS
Mul NUMBER;
BEGIN
    FOR  I  IN  1. .b
    LOOP
        Mul : = a * I;
        DBMS_OUTPUT.PUT_LINE (a || ‘*” || I || ‘=’ || Mul);
    END LOOP;
END;
```

Q.13 Consider the EMPLOYEE (EMPNO, SALARY, ENAME) Table.

Write a procedure raise_sal which increases the salary of an employee. It accepts an employee number and salary increase amount. It uses the employee number to find the current salary from the EMPLOYEE table and update the salary.

Ans

```
CREATE OR REPLACE PROCEDURE raise_sal
(       Mempno  EMPLOYEE . EMPNO % TYPE,
        Msal_percent   NUMBER )             AS
        Msal  EMPLOYEE . SALARY%TYPE;
BEGIN
UPDATE EMPLOYEE SET SALARY = SALARY + SALARY*Msal_percent /100
WHERE EMPNO = Mempno;
END;
```

Q.14 Write a PL/SQL function **CheckDiv that takes two numbers as arguments and returns the values 1 if the first argument passed to it is divisible by the second argument, else will return the value 0;**

Ans.

```
CREATE OR REPLACE FUNCTION CheckDiv (N1 NUMBER, N2 NUMBER) RETURN  NUMBER
AS
RES NUMBER;
BEGIN
```

```
            IF MOD (N1, N2) = 0 THEN
                        RES := 1;
            ELSE
                        RES:=  0;
            END IF;
                        RETURN RES;
END;
```

Q.15 Write a PL/SQL function called POW that takes two numbers as argument and return the value of the first number raised to the power of the second .

Ans.

```
CREATE OR REPLACE FUNCTION  POW (N1  NUMBER, N2 NUMBER) AS
RETURN  NUMBER
AS
RES NUMBER;
BEGIN
            SELECT POWER ( N1, N2) INTO RES FROM DUAL;
            RETURN RES;
END;
```

OR

```
CREATE OR REPLACE FUNCTION  POW (N1  NUMBER, N2 NUMBER) AS
RETURN  NUMBER
AS
            RES NUMBER : =1;
BEGIN
            FOR  RES  IN  1. .N2
            LOOP
                        RES : = N1 * RES;
            END LOOP;
            RETURN RES;
END;
```

Q.16 Write a PL/SQL function ODDEVEN to return value TRUE if the number passed to it is EVEN else will return FALSE.

Ans

```
CREATE OR REPLACE FUNCTION ODDEVEN (N NUMBER)
RETURN  BOOLEAN
AS
BEGIN
IF MOD (N, 2) = 0 THEN
            RETURN TRUE;
ELSE
            RETURN FALSE;
END IF;
END;
```

Q.17 Write a PL/SQL procedure EDSAL to find out whether the salary of an employee with ID =1234 is less than 5000 or not. If it is less than 5000, modify the salary of employee by increasing it by 15%. The table is - EMPLOYEE(ID, FIRST_NAME, LAST_NAME, EMAIL_ID, SALARY)

Ans

```
CREATE OR REPLACE PROCEDURE EDSAL
```

```
AS
    Msal  EMPLOYEE . SALARY%TYPE;
BEGIN
    SELECT SALARY INTO Msal FROM EMPLOYEE WHERE ID = 1234;
    IF Msal < 5000 THEN
        UPDATE EMPLOYEE SET SALARY = SALARY + SALARY*0.15
        WHERE ID = 1234;
    END IF;
END;
```

Q.18 Write a PL/SQL function called MYADDITION that takes two numbers as argument and return the sum of both values.

Ans

```
CREATE OR REPLACE FUNCTION MYADDITION (N1  NUMBER, N2 NUMBER) AS
RETURN  NUMBER
AS
BEGIN
    RETURN N1+N2;
END;
```