Practical: 05 Functions

A function is similar to a procedure except that a function must return a value.

A function is used for calculation purpose.

We can pass parameters to functions in three ways.

1) IN-parameters:

   The parameter can be referenced by the procedure or function. The value of the Parameter cannot be overwritten by the procedure or function.

2) OUT-parameters:

   An OUT parameter returns a value to the calling program. Inside the function, an OUT parameter acts like a variable. You can change its value And reference the value after assigning it. The actual parameter must be Variable and it is passed by value.

3) IN OUT-parameters:

   IN OUT parameter passes an initial value to a function and returns an updated value to the caller. It can be assigned a value and its value can be read.The actual parameter corresponding to an IN OUT formal parameter must be a variable, not a constant or an expression.

<u>Syntax:</u>

The simplified syntax for the CREATE FUNCTION statement is as follows:

```
CREATE [OR REPLACE] FUNCTION function_name
[(parameter_name [IN | OUT | IN OUT] type [, ...])]
RETURN type
{IS | AS}
BEGIN
      function_body
END function_name;
```

<u>Example:</u>

```
create or replace function getArea (i_rad NUMBER)
return NUMBER

 is

       v_pi NUMBER:=3.14;

begin

        return v_pi * (i_rad ** 2);

end;
```

<u>Execution of Function:</u>

```
SQL> exec dbms_output.put_line(getArea(2));

12.56
```

OR

```
SQL> select getArea(2) from dual;

GETAREA(2)

----------

12.56
```

<u>Create or Execute Function inside pl/sql Block:</u>

```
DECLARE
    r number(3);
    a number(3);
   function getArea (i_rad NUMBER)  return NUMBER
            is
                 v_pi NUMBER:=3.14;
    begin
                      return v_pi * (i_rad ** 2);
    end getArea;
 BEGIN
            r := &n1;
           a := getArea(r);
    dbms_output.put_line('Area of Circle is'||a);
 end;
 /
```

<u>Function Example with Queries:</u>

```
CREATE OR REPLACE FUNCTION totalemp
RETURN number IS
```

```
   total number(2) := 0;
BEGIN
   SELECT count(*) into total
   FROM employee;
   RETURN total;
END;
```

Execution of function in pl/sql Code:

```
DECLARE
       c number(2);
BEGIN
c := totalemp();
dbms_output.put_line('Total no. of Customers: ' || c);
END;
   Total no. of Customers: 2
```

Recursive Function Example:

```
 DECLARE
     num number;
     factorial number;
             FUNCTION fact(x number)
             RETURN number
             IS
                   f number;
     BEGIN
            IF x=0 THEN
               f := 1;
           ELSE
          f := x * fact(x-1);
          END IF;
                 RETURN f;
            END;
BEGIN
     num:= 6;
     factorial := fact(num);
     dbms_output.put_line(' Factorial '|| num || ' is ' ||
factorial);
END;
  /
Factorial 6 is 720
PL/SQL procedure successfully completed.
```

DROP a standalone function:

```
DROP FUNCTION function_name;
```

Exercise:

Employee(empid,ename,hire-date,designation,deptid)

1	Create a function which calculates average salary of department where employee is working and returns value of average salary to function. Employee id is passed as a parameter to the function.
2	Create a pl/sql block with function cmd_int in which pass parameters amount,rate,interest. Calculate compound interest or simple interest. Allow the user to select either simple interest or compound interest using case. Function returns value of result and which is displayed to the user.
3	Create a function called ***cal_experience*** which calculates the experience of employee whose empid is passed as a parameter into the function.(if month difference is –ve subtract 1 from total experience and then return the value) Write a pl/sql block to execute the function. Hint: 1) use inbuilt date functions
4	Create a pl/sql block which has Prime_number() function which takes one no as input. Function returns true if no is prime, and false otherwise.
5	Create a function palindrome() which pass a number or a text as input. Check if the input is a palindrome or not and return either true or false **Hint:** Use the TO_NUMBER() function to convert charcter to number Use SUBSTRING(str,pos) function to get part of string