**EX.NO. 8 PL/SQL FUNCTIONS**

**DATE:**

**AIM**:

To implement and execute functions in Oracle database using PL/SQL functions.

**PROCEDURE**:

STEP 1: Initialize the necessary parameters.

STEP 2: Develop the set of statements with the essential operational parameters.

STEP 3: Specify the Individual operation to be carried out.

STEP 4: Execute the function.

**FUNCTIUONS**:

Function is a subprogram that computes and returns a single value Functions and procedures are structured alike, except that functions have a RETURN clause. A function has two parts: the specification and the body. The function specification begins with the keyword FUNCTION and ends with the RETURN clause, which specifies the data type of the result value. Parameter declarations are optional. Functions that take no parameters are written without parentheses. The function body begins with the keyword IS and ends with the keyword END followed by an optional function name.

**Syntax:**

**CREATE**[OR REPLACE] FUNCTION functionname  
 [(parametername [IN | OUT | IN OUT] type [, ...])]  
 RETURN type  
 {IS | AS}  
 **BEGIN**  
   function\_body  
 **END**function\_name;

Where

1. OR REPLACE specifies the function that is to replace an existing function if presents
2. Type specifies the PL/SQL type of the parameter
3. The body of a function must return a value of the PL/SQL type specified in the RETURN Clause
4. FIND MAXIMUM
5. 4. SQL> declare
6. 5. 2 a number;
7. 6. 3 b number;
8. 7. 4 c number;
9. 8. 5 function findmax(x in number,y in number)
10. 9. 6 return number
11. 10. 7 is
12. 11. 8 z number;
13. 12. 9 begin
14. 13. 10 if x>y then
15. 14. 11 z:=x;
16. 15. 12 else
17. 16. 13 z:=y;
18. 17. 14 end if;
19. 18. 15 return z;
20. 19. 16 end;
21. 20. 17 begin
22. 21. 18 a:=23;
23. 22. 19 b:=45;
24. 23. 20 c:=findmax(a,b);
25. 24. 21 dbms\_output.put\_line('maximum of(23,45):'||c);
26. 25. 22 end;
27. 26. 23 /
28. OUTPUT
29. maximum of(23,45):45
30. PL/SQL procedure successfully completed.
31. FACTORIAL
32. 2 num number;
33. 3 factorial number;
34. 4 function fact(x number)
35. 5 return number
36. 6 is
37. 7 f number;
38. 8 begin
39. 9 if x=0 then
40. 10 f:=1;
41. 11 else
42. 12 f:=x\*fact(x-1);
43. 13 end if;
44. 14 return f;
45. 15 end;
46. 16 begin
47. 17 num:=6;
48. 18 factorial:=fact(num);
49. 19 dbms\_output.put\_line('factorial num 6 is :'||factorial);
50. 20 end;
51. 21 /
52. OUTPUT
53. factorial num 6 is :720
54. PL/SQL procedure successfully completed.
55. create function findaddress\_6(phone in number)
56. return varchar2 as address varchar2(100);
57. begin
58. select name||','||doorno','||','||place','||into address from emp\_6 where phoneno:=phoneno;
59. return address; exception when no\_data\_found then
60. return address not found;
61. end;
62. PL/SQL procedure successfully completed.
63. DISPLAY DETAILS
64. SQL> create or replace function disp(phone in number)
65. 2 return varchar2 as address varchar2(100);
66. 3 begin
67. 4 select name||','||doorno||','||place into address from emp\_6 where phoneno=phone;
68. 5 return address;
69. 6 exception when no\_data\_found then return 'address not found';
70. 7 end;
71. 8 /
72. Function created.
73. SQL> declare
74. 2 address varchar2(100);
75. 3 begin
76. 4 address:=disp(546587);
77. 5 dbms\_output.put\_line(address);
78. 6 end;
79. 7 /
80. OUTPUT
81. ragu,7,namakkal
82. PL/SQL procedure successfully completed.