Oracle Function

A function is a subprogram that is used to return a single value. You must declare and define a function before invoking it. It can be declared and defined at a same time or can be declared first and defined later in the same block.

What is a Function in Oracle?

A function is a named PL/SQL block that returns a value (In general to compute A value) that can be stored in the database as a schema object for repeated execution.

Thisis called a part of an expression or is used to provide a parameter value. We can use the part of the function as an expression. We can create a function and call the function at any time. This is the difference between procedure and function. The syntax of the function is very simple and easy. The PL/SQL block must have at least one RETURN statement.

Examples of Functions:

- A function can be created to return the salary of an employee.
- Function to retrieve the full name of the employee.
- Function to calculate the GPA for the student.
- Function to compute the tax of the salary.

Differences Between Functions and Procedures in Oracle

Procedures	Functions
Execute a PL/SQL statement	Invoke as a part of an expression.
Does not contain a RETURN clause in the header.	Must contain a RETURN clause in the header.
Can pass values using output parameters	Must return a single value.
Can contain a RETURN statement without a value.	Must contain at least one RETURN statement.
To Perform an action	To return a value.
Cannot be used in select	Can be used in the select statements, but it should not

statements.	include OUT/ IN OUT parameters.

```
create or replace function get_sal
(p_emp_id number)
return number
is
v_sal number;
begin
  select salary into v_sal
  from Employee
  where Id=p_emp_id;
  return v_sal;
end;

//--------
declare
v_sal number;
begin
v_sal:=get_sal(100);
dbms_output.put_line (v_sal);
end;
//---------
begin
dbms_output.put_line (get_sal(100));
end;
//--------------------
execute dbms_output.put_line(get_sal(100));

//-------
VARIABLE B_SAL NUMBER
EXECUTE :B_SAL:=get_sal(100);
PRINT  B_SAL;


select * from user_objects where object_name='GET_SAL';
select line, text from user_source where name='GET_SAL';




create or replace function get_sal
(p_emp_id number)
return number
is
v_sal number;
begin
  select salary into v_sal
  from Employee
  where Id=p_emp_id;
  return v_sal;

  return v_sal;
  exception
  when no_data_found then
  return -1;
```

```
end;
```

```
create or replace function get_sal_tax
(p_sal number)
return number
is

begin

 if p_sal<50000 then
 return p_sal* (10/100);
 else
 return p_sal* (15/100);
 end if;

end;









Select id,name,salary, get_sal_tax(salary) from Employee;

Select id,name,salary, get_sal_tax(p_sal=salary) from Employee;

Select id, name, salary, get_sal_tax(salary) from Employee
where get_sal_tax(salary)>2000 order by get_sal_tax(salary);







Select id, name, salary, get_sal_tax(salary) from Employee
where get_sal_tax(salary)>2000 and order by get_sal_tax(salary);
```

CREATE function

```
create or replace function addition(n1 in number, n2 in number)
return number
is
n3 number(8);
```

```
begin
n3 :=n1+n2;
return n3;
end;
DECLARE
   n3 number(2);
BEGIN
   n3 := addition(11,22);
   dbms_output.put_line('Addition is: ' || n3);
END;
```

```
DECLARE
       a number;
       b number;
       c number;
    FUNCTION findMax(x IN number, y IN number)
    RETURN number
    IS
        z number;
    BEGIN
       IF x > y THEN
          z:= x;
       ELSE
          Z:= y;
       END IF;

       RETURN z;
    END;
    BEGIN
       a:= 23;
       b:= 45;

       c := findMax(a, b);
       dbms_output.put_line(' Maximum of (23,45): ' || c);
    END;




CREATE OR REPLACE FUNCTION Total_Emp
RETURN number IS
   total number(2) := 0;
BEGIN
   SELECT count(*) into total
   FROM employee;
    RETURN total;
END;

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

```
DROP FUNCTION function_name;
```