

### Syntax to Create a Function

```
CREATE OR REPLACE FUNCTION function_name [parameters]
    RETURN return datatype
{IS | AS}
BEGIN
    < function_body >
END function_name;
```

Q. Write a function which accept two numbers and return its addition

```
Create or Replace Function addFunction(
n1 IN number, n2 IN Number)
Return number
IS
    n3 number;
Begin
    n3:=n1+n2;
    return n3;
End;
```

### How to call Function?

```
declare
n number;
Begin
    n:=addFunction(10,20);
    dbms_output.put_line('Addition is '||n);
End;
```

Q. Write a function which accept two numbers and return its addition

```
Create or Replace Function addFunction(  
n1 IN number, n2 IN Number)  
Return number  
IS  
    n3 number;  
Begin  
    n3:=n1+n2;  
    return n3;  
End;
```

Q. Write a function which accept Employee number and return employee salary

---

```
Create Or Replace Function GetSalary(  
v_empno IN Number)  
Return Number  
AS  
    v_salary number;  
Begin  
    select salary into v_salary From Employee101  
    where emp_no=v_empno;  
    return v_salary;  
    Exception when others then  
        v_salary:=0;  
        return v_salary;  
        dbms_output.put_line('Exception has Occured');  
End;
```

Q. Write a Function which accept employee number and return employee name associated with that Employee number

```
Create or Replace Function EmpName(  
v_empno IN Number)  
Return Employee.empname%Type  
IS  
    v_empname varchar(250);  
Begin  
    select empname into v_empname From Employee  
    where empno=v_empno;  
    return v_empname;  
END EmpName;
```

#### How to Call?

```
declare  
v_empname varchar(250);  
Begin  
    v_empname:=EmpName(101);  
    dbms_output.put_line('Employee Name is '||v_empname);  
  
End;
```

#### # Difference between Function and Procedure

- 1) Procedure May or may not return a value but function always return a value.
- 2) Function can call Procedure but Procedure cannot call FUNCTION
- 3) DML statments cannot be executed within a function  
DML statements can be executed within a procedure.
- 4) In select statement we can use function but  
in select statement we cannot use Procedure

Demo Program which shows DML statement not allowed in function

```
Create or Replace Function EmpName(  
v_empno IN Number,v_empname IN Employee.empname%Type,  
v_city IN Employee.city%Type)  
Return Employee.empname%Type  
IS  
    p_empname varchar(250);  
Begin  
    --Insert into Employee values(v_empno,v_empname,v_city);  
    --commit;  
    select empname into p_empname From Employee  
    where empno=v_empno;  
    return p_empname;  
END EmpName;
```

### Assignment Questions

**Q. Write a function to count number of Employee**

**Q. Write a function to find max salary**

**Q. Write a function to find min salary**

**Q. Write a function to avg salary**

What is mean by Trigger?

-> Triggers are stored programs which get executed automatically or get fired automatically when some events occurs.  
like

DML statement -DELETE, INSERT, or UPDATE

DDL Statement -CREATE, ALTER, or DROP

Database Operations-SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN

-> Triggers can be defined on the table, view, schema, or database with which the event is associated.

### -> How to Create a Trigger

```
CREATE OR REPLACE TRIGGER trigger_name
BEFORE | AFTER
INSERT [OR] | UPDATE [OR] | DELETE
|CREATE |[OR] |ALTER [OR] | DROP
ON table_name
[REFERENCING OLD AS o NEW AS n]
[FOR EACH ROW]
WHEN (condition)
DECLARE
    Declaration-statements
BEGIN
    Executable-statements
EXCEPTION
    Exception-handling-statements
END;
```

Words Meaning which are used in Trigger Syntax

**BEFORE | AFTER** -> When we want to fire a Trigger

**INSERT | UPDATE | DELETE**

**CREATE |ALTER |DROP** ---> All these are events

**Q. Consider Student Table , When we are updating student marks and when new marks are less than old mark then fire a Trigger**

```
Create Or Replace Trigger Update_student_Mark
Before
Update
ON Student301
For Each Row
Begin
IF (:NEW.mark< :OLD.mark) Then
raise_application_error(-20000,'New Marks '||:NEW.mark ||'
must be greater than Old Mark '||:OLD.mark);
End IF;
End;
```

**Now Execute the Update statement**

```
update Student301 set mark=25 where rollno=101;
```

Example 2:

```
Create Or Replace Trigger Update_student_Mark
Before
Update OR Insert OR Delete
ON Student301
For Each Row
Begin
IF (:NEW.mark< :OLD.mark) Then
raise_application_error(-20000,'New Marks '||:NEW.mark ||'
must be greater than Old Mark '||:OLD.mark);
End IF;

IF (:NEW.mark>100) Then
raise_application_error(-20000,'New Marks '||:NEW.mark ||'
Should be less or Equal to 100');
End IF;
End;
```

about `raise_application_error()` function

- > It is predefined function used to raise an exception manually
- > It accept two parameter
  - i) Error Number
  - ii) Error Message
- > Allowed range for Error Number is **-20000 to -20999.**

1) How Enable or Disable Trigger

-> To disable Trigger below syntax is used

```
ALTER TRIGGER trigger_name DISABLE;
```

2) How Enable the Trigger ?

-> To Enable the Trigger below syntax is used

```
ALTER TRIGGER trigger_name ENABLE;
```

3) How to Drop Trigger?

-> To Drop trigger below syntax is used?

```
Drop Trigger trigger_name;
```

### Assignment Questions

- 1) Write a trigger which does not allow to insert or Update the Employee name whose length is greater than 10

```
Create Or Replace Trigger Update_Employee
Before
Update OR Insert
ON Employee
For Each Row
Declare
n number;
Begin
select length(:NEW.empname) into n From dual;
IF (n>10) Then
    raise_application_error(-20000,'Lenght shouldb be less or equal to 10');
End IF;
End;
```

```
insert into Employee values(106,'Abxacvbgsdfgh','Pune','IT');
```