

```
SET SERVEROUTPUT ON;  
SET VERIFY OFF;
```

```
DECLARE
```

```
X number:= &Fund_Amount_to_add;
```

```
v_fund_amnt number;
```

```
Begin
```

```
    select fund_amnt into v_fund_amnt from fund where fid=1;  
    dbms_output.put_line('Old Fund: '||v_fund_amnt);  
    v_fund_amnt:= v_fund_amnt+X;  
    dbms_output.put_line('New Fund: '||v_fund_amnt);  
    update FUND set fund_amnt=v_fund_amnt where fid=1;  
    commit;
```

```
EXCEPTION
```

```
When no_data_found then
```

```
    DBMS_OUTPUT.PUT_LINE('No Data Found');
```

```
When others then
```

```
    DBMS_OUTPUT.PUT_LINE('Something wrong happened');
```

```
end;
```

```
/
```

```
clear screen;
SET SERVEROUTPUT ON;
SET VERIFY OFF;
```

```
DECLARE
```

```
    v_lid LEAVE.lid %type;
    v_eid LEAVE.eid %type:=&EmployeeID;
    v_l_month LEAVE.l_month %type:='&Month';
    v_l_day LEAVE.l_day %type:=&No_of_Leaves;
```

```
BEGIN
```

```
    SELECT lid
    into v_lid
    FROM (select * from LEAVE ORDER BY lid DESC) leave1
    WHERE rownum <= 1 ORDER BY rownum DESC;
    v_lid:=v_lid+1;
    AddLeave(v_lid,v_eid,v_l_month,v_l_day);
```

```
    EXCEPTION
```

```
    When no_data_found then
```

```
        DBMS_OUTPUT.PUT_LINE('No Data Found');
```

```
    When others then
```

```
        DBMS_OUTPUT.PUT_LINE('Something wrong happened');
```

```
END;
```

```
/
```

```
clear screen;  
SET SERVEROUTPUT ON;  
SET VERIFY OFF;
```

```
DECLARE
```

```
X number:= &Eid;  
Y varchar2(15):= '&Month_of_Salary';
```

```
r number;
```

```
Begin
```

```
  r := myPackage.Generate_Salary(X,Y);
```

```
  DBMS_OUTPUT.PUT_LINE(r);  
  myPackage.Transect_Salary(X,r,Y);  
  commit;  
end;  
/
```

```
clear screen;
SET SERVEROUTPUT ON;
SET VERIFY OFF;
```

```
DECLARE
```

```
    v_eid EMPLOYEE.eid %TYPE;
    v_ename EMPLOYEE.ename %TYPE :='&Name';
    v_gender EMPLOYEE.gender %TYPE :='&Gender';
    v_email EMPLOYEE.email %TYPE :='&Email';
    v_joinDate EMPLOYEE.join_date %TYPE:='&JoinDate';
    v_sid employee_salary.sid%TYPE:='&Salary';
```

```
BEGIN
```

```
    SELECT eid
    into v_eid
    FROM (select * from EMPLOYEE ORDER BY eid DESC) Emp1
    WHERE rownum <= 1 ORDER BY rownum DESC;
    v_eid:=v_eid+1;
    insert into employee values (v_eid, v_ename, v_gender,v_email, v_joinDate);
    insert into employee_salary values(v_eid,v_sid);
    commit;
```

```
EXCEPTION
```

```
When no_data_found then
```

```
    DBMS_OUTPUT.PUT_LINE('No Data Found');
```

```
When others then
```

```
    DBMS_OUTPUT.PUT_LINE('Something wrong happened');
```

```
END;
```

```
/
```

```
drop database link site_link;
```

```
create database link site_link  
connect to system identified by "123456"  
using '(DESCRIPTION =  
  (ADDRESS_LIST =  
    (ADDRESS = (PROTOCOL = TCP)  
      (HOST = 192.168.56.1)  
      (PORT = 1521))  
  )  
  (CONNECT_DATA =  
    (SID = XE)  
  )  
)'  
;
```

```
clear screen;
SET SERVEROUTPUT ON;
SET VERIFY OFF;

DECLARE
    v_eid EMPLOYEE.eid %TYPE:=&EmpolyeelD;
    v_sid employee_salary.sid%TYPE:=&NewSalaryId;

BEGIN
    ChangeEmployeeSalary(v_eid,v_sid);
END;
/
```

SET SERVEROUTPUT ON;

SET VERIFY OFF;

clear screen;

DECLARE

X number:= &Your_EmployeeID;

--v_eid EMPLOYEE.eid %TYPE;

v_ename EMPLOYEE.ename %TYPE;

v_gender EMPLOYEE.gender %TYPE;

v_email EMPLOYEE.email %TYPE;

v_joinDate EMPLOYEE.join_date %TYPE;

v_l_day leave.l_day%TYPE;

v_basic salary.basic%TYPE;

v_allowance salary.allowance%TYPE;

v_sid employee_salary.sid%TYPE;

n1 number;

v_tid transection.tid %TYPE;

v_eid transection.eid %TYPE;

v_ammount transection.ammount %TYPE;

v_t_date transection.t_date %TYPE;

v_s_month transection.s_month %TYPE;

--s1 varchar2;

cursor payments_cur is

select tid, eid, s_month,t_date,ammount from TRANSECTION@site_link where eid=X;

Begin

select ename,gender,email,join_date

into v_ename,v_gender,v_email,v_joinDate

from EMPLOYEE@site_link where eid=X;

select sid into v_sid from employee_salary@site_link where eid=X;

select basic,allowance into v_basic,v_allowance from salary@site_link where

sid=v_sid;

dbms_output.put_line('_____');

dbms_output.put_line('Employee Details_____');

dbms_output.put_line('Employee ID :'||X);

dbms_output.put_line('Name :'||v_ename);

dbms_output.put_line('Gender :'||v_gender);

dbms_output.put_line('Email :'||v_email);

dbms_output.put_line('Joining Date :'||v_joinDate);

dbms_output.put_line('_____');

```

        dbms_output.put_line('Salary Details_____');
        dbms_output.put_line('Basic      : '||v_basic);
        dbms_output.put_line('Allowance   : '||v_allowance);
        n1:=v_basic+v_allowance;
        dbms_output.put_line('Total Salary : '||n1);
        dbms_output.put_line('_____');
        dbms_output.put_line('Payments Details_____');
        dbms_output.put_line('TransectionID EmployeeID MonthOfSalary Leaves
PaymentDate Amount');
        open payments_cur;

        loop
            fetch payments_cur into v_tid,v_eid,v_s_month,v_t_date,v_ammount;
            exit when payments_cur%notfound;
            select l_day into v_l_day from leave@site_link where eid=X and
l_month=v_s_month;
            DBMS_OUTPUT.PUT_LINE(TO_CHAR(v_tid)||'          '||TO_CHAR(v_eid)||'
'||TO_CHAR(v_s_month)||'      '||TO_CHAR(v_l_day)||'      '||TO_CHAR(v_t_date)||'
'||TO_CHAR(v_ammount));
            end loop;
            close payments_cur;

EXCEPTION
    WHEN no_data_found THEN
        dbms_output.put_line('THIS EMPLOYEE DOESNT EXIST!');
    WHEN others THEN
        dbms_output.put_line('ERROR!');
end;
/

```



```

create or replace function Check_Valid(n1 in number, s1 in varchar2)
return number
is

    temp_eid transection.eid %TYPE:= n1;
    temp_s_month transection.s_month %TYPE:=s1;

    v_eid transection.eid %TYPE;
    v_s_month transection.s_month %TYPE;

    return_val number:=0;

    cursor my_cur is
    select eid, s_month from TRANSECTION;

BEGIN

    OPEN my_cur;
        loop
            fetch my_cur into v_eid,v_s_month;
            exit when (my_cur%notfound or return_val=1) ;
            if((v_eid=temp_eid) and (v_s_month=temp_s_month)) then
                return_val:=1;
            else
                return_val:=2;
            end if;
        end loop;
    CLOSE my_cur;

    return return_val;

EXCEPTION
When no_data_found then
    DBMS_OUTPUT.PUT_LINE('No Data Found');
When others then
    DBMS_OUTPUT.PUT_LINE('Something wrong happened');

end Check_Valid;
/

```

create or replace package myPackage as

 function Generate_Salary(n1 in number, s1 in varchar2)
 return number;

 Procedure Transect_Salary(n1 in number, n2 in number,s1 in varchar2);

end myPackage;

/

```
create or replace Procedure AddLeave(v_lid in number, v_eid in number,v_l_month in
varchar2,v_l_day in number)
is
begin
insert into leave values(v_lid,v_eid, v_l_month, v_l_day);
commit;
end AddLeave;
/
```

```
create or replace Procedure ChangeEmployeeSalary(v_eid in number,v_sid in number)
is
Begin

update EMPLOYEE_SALARY set sid=v_sid where eid=v_eid;
commit;
EXCEPTION
    When no_data_found then
        DBMS_OUTPUT.PUT_LINE('No Data Found');
    When others then
        DBMS_OUTPUT.PUT_LINE('Something wrong happened');

end ChangeEmployeeSalary;
/
```

```
create or replace Procedure Update_Fund(n1 in number)

    is
        v_fund_amnt fund.fund_amnt %TYPE;
Begin

select fund_amnt into v_fund_amnt from fund where fid=1;
v_fund_amnt:= v_fund_amnt-n1;

update FUND set fund_amnt=v_fund_amnt where fid=1;
commit;

EXCEPTION
    When no_data_found then
        DBMS_OUTPUT.PUT_LINE('No Data Found');
    When others then
        DBMS_OUTPUT.PUT_LINE('Something wrong happened');

end Update_Fund;
/
```

```
SET SERVEROUTPUT ON;
```

```
create or replace trigger Trig_AddLeave
```

```
after insert on Leave
```

```
begin
```

```
    dbms_output.put_line('Employee leaves added according to month');
```

```
end;
```

```
/
```

```
SET SERVEROUTPUT ON;
```

```
CREATE OR REPLACE trigger EmpSalary_audit
```

```
BEFORE UPDATE ON EMPLOYEE_SALARY
```

```
FOR EACH ROW
```

```
DECLARE
```

```
    v_date varchar2(30);
```

```
begin
```

```
    select sysdate into v_date from dual;
```

```
    insert into EMPLOYEE_SALARY_Audit values (:NEW.sid,:OLD.sid,v_date);
```

```
    dbms_output.put_line('Salary Change for The Employee');
```

```
    commit;
```

```
end;
```

```
/
```

```
SET SERVEROUTPUT ON;
```

```
create or replace trigger Trig_EmpSalaryInsert
```

```
after insert on EMPLOYEE_SALARY
```

```
begin
```

```
    dbms_output.put_line('Salary Assigned To the Employee');
```

```
end;
```

```
/
```



```
SET SERVEROUTPUT ON;
```

```
create or replace trigger Trig_EmployeeInsert  
after insert on Employee  
begin  
    dbms_output.put_line('One Employee Added');  
end;  
/
```

```
SET SERVEROUTPUT ON;
```

```
create or replace trigger Trig_Tansect
```

```
after insert on transection
```

```
begin
```

```
    dbms_output.put_line('One Transection completed');
```

```
end;
```

```
/
```

```
SET SERVEROUTPUT ON;
```

```
CREATE OR REPLACE trigger Fund_audit
```

```
BEFORE UPDATE ON FUND
```

```
FOR EACH ROW
```

```
DECLARE
```

```
    v_date varchar2(30);
```

```
begin
```

```
    select sysdate into v_date from dual;
```

```
    insert into FUND_Audit values (:NEW.fund_amnt,:OLD.fund_amnt,v_date);
```

```
    dbms_output.put_line('Fund Updated');
```

```
    commit;
```

```
end;
```

```
/
```

```

create or replace package body myPackage as
    function Generate_Salary(n1 in number, s1 in varchar2)
    return number
    is

        v_l_day leave.l_day%TYPE;
        v_basic salary.basic%TYPE;
        v_allowance salary.allowance%TYPE;
        v_sid employee_salary.sid%TYPE;
        v_return number;

begin

    select l_day into v_l_day from leave where eid=n1 and l_month=s1;
    select sid into v_sid from employee_salary where eid=n1;
    select basic,allowance into v_basic,v_allowance from salary where sid=v_sid;

    if(v_l_day<3) then
        v_return:=v_basic+v_allowance;

    elsif(v_l_day>=3 and v_l_day<5) then
        v_basic:=v_basic-(v_basic*0.05);
        v_return:=v_basic+v_allowance;

    elsif(v_l_day>=5) then
        v_basic:=v_basic-(v_basic*0.07);
        v_return:=v_basic+v_allowance;

    end if;
    return v_return;

EXCEPTION
    When no_data_found then
        DBMS_OUTPUT.PUT_LINE('No Data Found');
    When others then
        DBMS_OUTPUT.PUT_LINE('Something wrong happened');

end Generate_Salary;

Procedure Transect_Salary(n1 in number, n2 in number,s1 in varchar2)
is

```

```

t_id transection.tid %TYPE;
v_count NUMBER:=0;
new_id NUMBER;

v_tid transection.tid %TYPE;
v_eid transection.eid %TYPE:= n1;
v_ammount transection.ammount %TYPE :=n2;
v_t_date transection.t_date %TYPE;
v_s_month transection.s_month %TYPE:=s1;

checking number      ;

BEGIN

select sysdate into v_t_date from dual;
select count(*) into t_id from transection where tid=1;
if(t_id=0) then
    insert into transection values(1,v_eid,v_ammount,v_t_date,v_s_month);

    Update_Fund(v_ammount);
else

    checking := Check_Valid(v_eid,v_s_month);
    if(checking=2) then

        SELECT tid
        into t_id
        FROM (select * from transection ORDER BY tid DESC) transection1
        WHERE rownum <= 1 ORDER BY rownum DESC;

        v_count:=v_count+1;
        new_id := t_id + v_count;
        insert into transection values(new_id,v_eid,v_ammount,v_t_date,v_s_month);
        Update_Fund(v_ammount);
    elsif (checking=1) then
        dbms_output.put_line('Already done');
    end if;
end if;
commit;

end Transect_Salary;

end myPackage;

```


create database smdb

clear screen;

```
DROP TABLE EMPLOYEE CASCADE CONSTRAINTS;  
DROP TABLE SALARY CASCADE CONSTRAINTS;  
DROP TABLE EMPLOYEE_SALARY CASCADE CONSTRAINTS;  
DROP TABLE TRANSECTION CASCADE CONSTRAINTS;  
DROP TABLE LEAVE CASCADE CONSTRAINTS;  
DROP TABLE FUND CASCADE CONSTRAINTS;
```

```
CREATE TABLE EMPLOYEE (  
    eid int,  
    ename varchar2(20),  
    gender varchar2(5),  
    email varchar2(20),  
    join_date varchar2(20) ,  
    PRIMARY KEY(eid));
```

```
CREATE TABLE SALARY(  
    sid int,  
    basic int,  
    allowance int,  
    PRIMARY KEY(sid));
```

```
CREATE TABLE EMPLOYEE_SALARY(  
    eid int,  
    sid int,  
    FOREIGN KEY(eid) REFERENCES EMPLOYEE(eid),  
    FOREIGN KEY(sid) REFERENCES SALARY(sid));
```

```
CREATE TABLE LEAVE(  
    lid int,  
    eid int,  
    l_month varchar2(15),  
    l_day int,  
    PRIMARY KEY(lid));
```

```
CREATE TABLE TRANSECTION (
```

```
tid int,  
eid int,  
ammount int,  
t_date date,  
s_month varchar2(15),  
PRIMARY KEY(tid),  
FOREIGN KEY(eid) REFERENCES EMPLOYEE(eid));
```

```
CREATE TABLE FUND (  
    fid int,  
    fund_amnt int,  
    PRIMARY KEY(fid));
```

```
CREATE TABLE FUND_Audit (  
    amnt_new int,  
    amnt_old int,  
    Update_date varchar2(30));
```

```
CREATE TABLE EMPLOYEE_SALARY_Audit(  
    new_sid int,  
    old_sid int,  
    Changing_date varchar2(30));
```

```
insert into employee values (1, 'Sajid Abdullah', 'M', 'sajid@gamil.com', '1/1/2019');  
insert into employee values (2, 'Samia Zahan', 'F', 'samia@gamil.com', '1/1/2019');  
insert into employee values (3, 'Muna Saha', 'F', 'muna@gamil.com', '1/1/2019');  
insert into employee values (4, 'Robiul Hasan', 'M', 'nowshad@gamil.com', '1/1/2019');  
insert into employee values (5, 'Shahriar Shibli', 'M', 'shibli@gamil.com', '1/1/2019');  
insert into employee values (6, 'Shourav Saha', 'M', 'shourav@gamil.com', '1/1/2019');  
insert into employee values (7, 'Apurbo Roy', 'M', 'apu@gamil.com', '1/1/2019');  
insert into employee values (8, 'Tanvir Rahman', 'M', 'mrt@gamil.com', '1/1/2019');  
insert into employee values (9, 'Sonia Khatun', 'F', 'sonia@gamil.com', '1/1/2019');  
insert into employee values (10, 'Humayan Nila', 'F', 'nila@gamil.com', '1/1/2019');
```

```
insert into salary values(1, 18000,5000);  
insert into salary values(2, 20000,5000);  
insert into salary values(3, 22000,6000);  
insert into salary values(4, 35000,6500);  
insert into salary values(5, 50000,7000);
```



```
insert into employee_salary values(1,1);
insert into employee_salary values(2, 3);
insert into employee_salary values(3,5);
insert into employee_salary values(4,2);
insert into employee_salary values(5,1);
insert into employee_salary values(6,3);
insert into employee_salary values(7,5);
insert into employee_salary values(8,4);
insert into employee_salary values(9,2);
insert into employee_salary values(10,4);
```

```
insert into leave values(1,1, 'Jan/19', 3);
insert into leave values(2,3, 'Jan/19', 4);
insert into leave values(3,2, 'Jan/19', 5);
insert into leave values(4,6, 'Jan/19', 3);
insert into leave values(5,4, 'Jan/19', 1);
insert into leave values(6,5, 'Jan/19', 5);
insert into leave values(7,9, 'Jan/19', 3);
insert into leave values(8,7, 'Jan/19', 3);
insert into leave values(9,8, 'Jan/19', 4);
insert into leave values(10,10, 'Jan/19', 3);
insert into leave values(11,1, 'Feb/19', 5);
```

```
insert into fund values(1,1000000);
```

```
commit;
```