**9. Write a procedure and a function.**

**Function: write a function to calculate number of years of experience of employee.(note:**

**pass hiredate as a parameter)**

**Procedure: Capture the value returned by the above function to calculate the additional**

**allowance for the emp based on the experience.**

**Additional Allowance = Year of experience x 3000**

**Calculate the additional allowance**

**and store Empno, ename,Date of Joining, and Experience in**

**years and additional allowance in Emp\_Allowance table.**

**create table emp\_allowance(**

**empno int,**

**ename varchar(20),**

**hiredate date,**

**experience int,**

**allowance decimal(9,2));**

delimiter //

create procedure question9()

begin

declare vfinished int default 0;

declare vempno,vdeptno int;

declare vename,vjob varchar(20);

declare vsal float(9,2);

declare vhiredate date;

declare vexp int;

declare vallowance float(9,2);

declare empcur1 cursor for select empno,ename,job,hiredate,sal,deptno from emp;

declare continue handler for NOT FOUND set vfinished=1;

open empcur1;

label1:loop

fetch empcur1 into vempno,vename,vjob,vhiredate,vsal,vdeptno;

if vfinished = 1 then

leave label1;

end if;

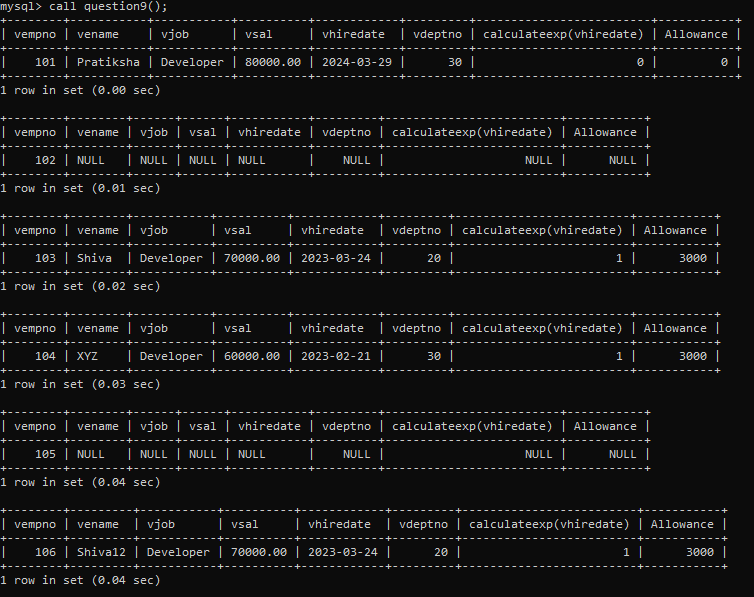
select vempno,vename,vjob,vsal,vhiredate,vdeptno,calculateexp(vhiredate),calculateexp(vhiredate)\*3000 As 'Allowance';

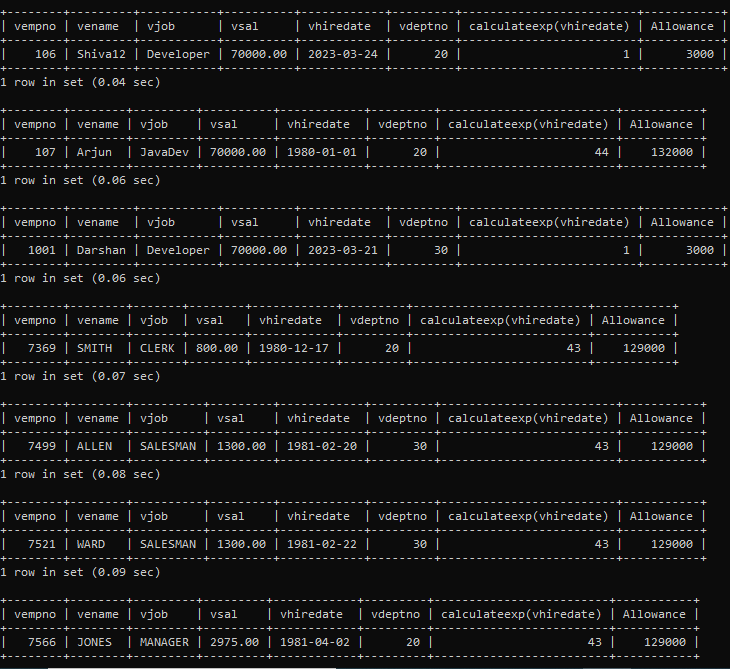
end loop;

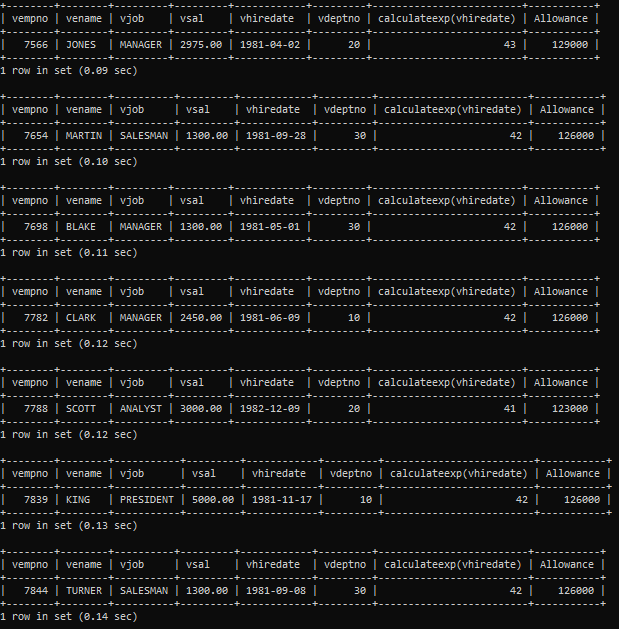
close empcur1;

end//

delimiter ;







**7. Write a procedure that displays the following information of all emp**

**Empno,Name,job,Salary,Status,deptno**

**Note: - Status will be (Greater, Lesser or Equal) respective to average salary of their own**

**department. Display an error message Emp table is empty if there is no matching**

**record.**

delimiter //

create procedure question7()

begin

declare vfinished int default 0;

declare vempno,vdeptno int;

declare vename,vjob,vstatus varchar(20);

declare vsal,vavgsal float(9,2);

declare vcnt int default 0;

declare empcur cursor for select empno,ename,job,sal,deptno from emp;

declare continue handler for NOT FOUND set vfinished=1;

open empcur;

label1:loop

fetch empcur into vempno,vename,vjob,vsal,vdeptno;

if vfinished = 1 then

if vcnt=0 then

select 'Emp table is empty';

end if;

leave label1;

end if;

set vcnt = vcnt+1;

select avg(sal) into vavgsal from emp

where deptno = vdeptno;

if vsal < vavgsal then

set vstatus ='lesser';

elseif vsal = vavgsal then

set vstatus ='Equal';

else

set vstatus ='Greater';

end if;

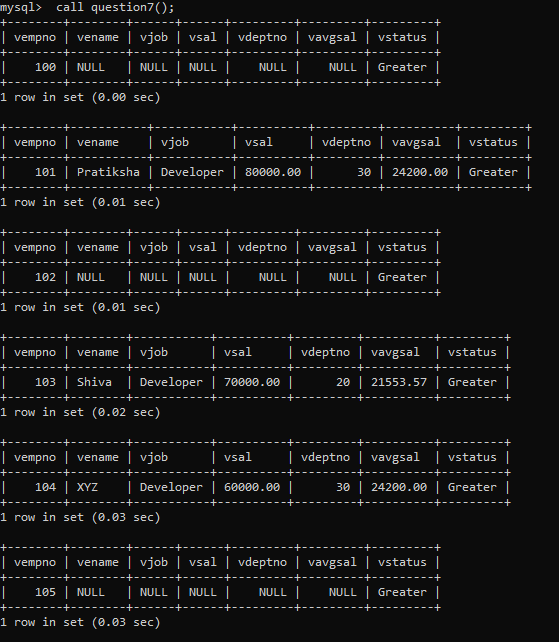
select vempno,vename,vjob,vsal,vdeptno,vavgsal,vstatus;

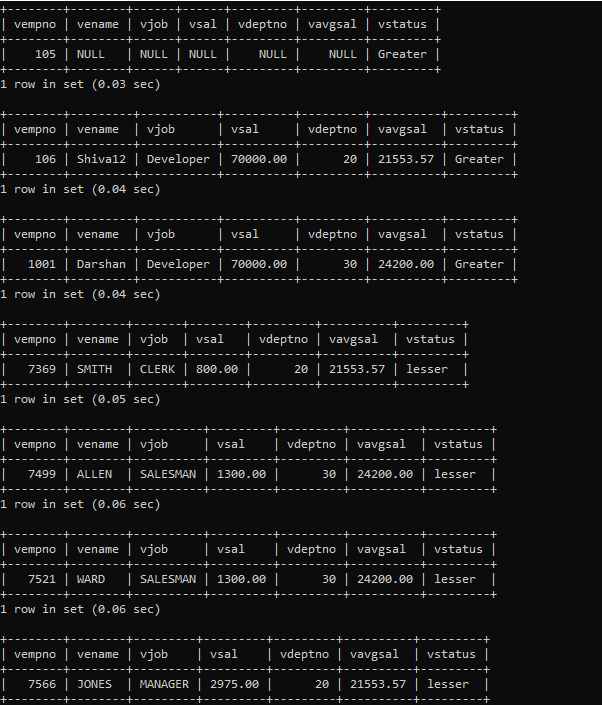
end loop;

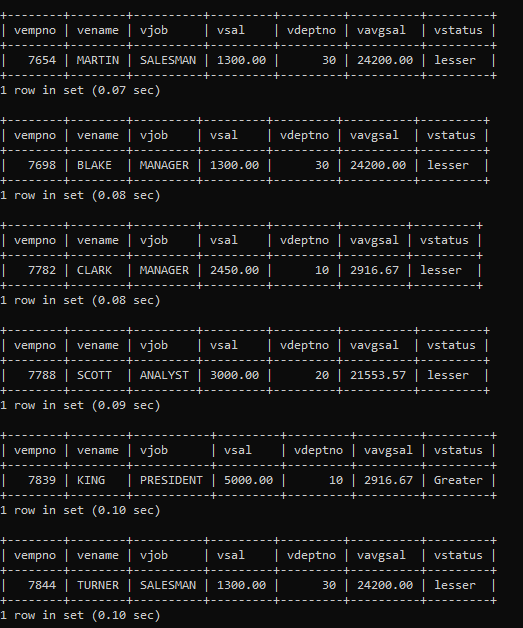
close empcur;

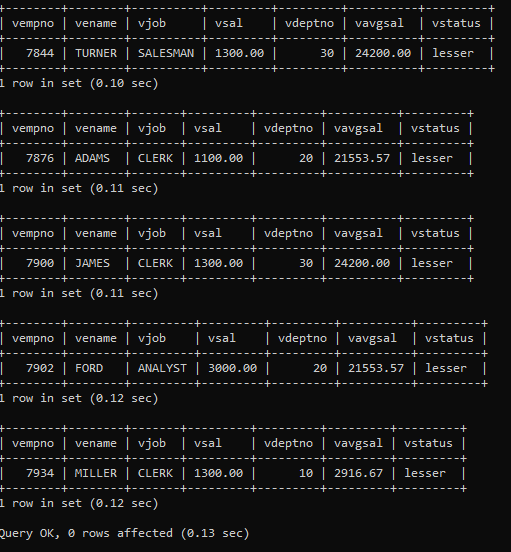
end//

delimiter ;









**8. Write a procedure to update salary in emp table based on following rules.**

**Exp< =35 then no Update**

**Exp> 35 and <=38 then 20% of salary**

**Exp> 38 then 25% of salary**

delimiter //

create procedure calculateexp123()

begin

declare vfinished int default 0;

declare vempno, vmgr,vdeptno,vexp int;

declare vename , vjob varchar(50);

declare vhiredate date;

declare vsal, vcomm ,vupdatedsal float(9,2);

declare empcur cursor for select \* from emp;

declare continue handler for NOT FOUND set vfinished=1;

open empcur;

label1:loop

fetch empcur into vempno,vename, vjob,vmgr,vhiredate,vsal,vcomm,vdeptno;

if vfinished=1 then

leave label1;

end if;

select floor(datediff(curdate(),vhiredate)/365),sal into vexp,vsal from emp where empno=vempno;

if vexp<=35 then

set vupdatedsal=vsal;

elseif vexp>35 and vexp<=38 then

update emp

set sal=vsal\*1.2

where empno=vempno;

set vupdatedsal=vsal\*1.2;

else

update emp

set sal=vsal\*1.25

where empno=vempno;

set vupdatedsal=vsal\*1.25;

end if;

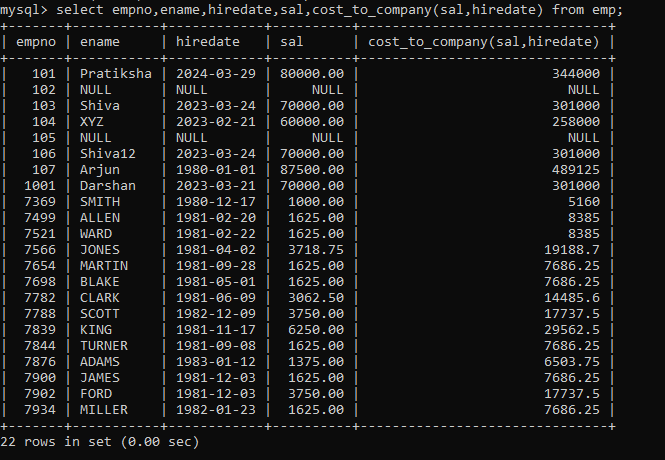
select vempno,vename,vjob,vsal,vhiredate,vdeptno,vexp;

end loop;

close empcur;

end //

delimiter ;



**10. Write a function to compute the following. Function should take sal and hiredate**

**as i/p and return the cost to company.**

**DA = 15% Salary, HRA= 20% of Salary, TA= 8% of Salary.**

**Special Allowance will be decided based on the service in the company.**

**< 1 Year Nil**

**>=1 Year< 2 Year 10% of Salary**

**>=2 Year< 4 Year 20% of Salary**

**>4 Year 30% of Salary**

**11. Write query to display empno,ename,sal,cost to company for all employees(note:**

**use function written in question 10)**

delimiter //

create function cost\_to\_company(esal float(9,2) ,shiredate date ) returns float

begin

declare DA,HRA,TA float(9,2);

declare specialA float(9,2);

declare vexp int;

declare vsal float(9,2);

set DA = esal\*(1.5);

set HRA = esal\*(2.0);

set TA = esal\*(0.8);

set vsal = DA+HRA+TA;

set vexp = floor(datediff(curdate(),shiredate)/365);

if vexp <= 1 then

set vsal = vsal;

elseif vexp <=42 then

set vsal = vsal+(vsal\*0.10);

elseif vexp <=43 then

set vsal = vsal+(vsal\*0.20);

else

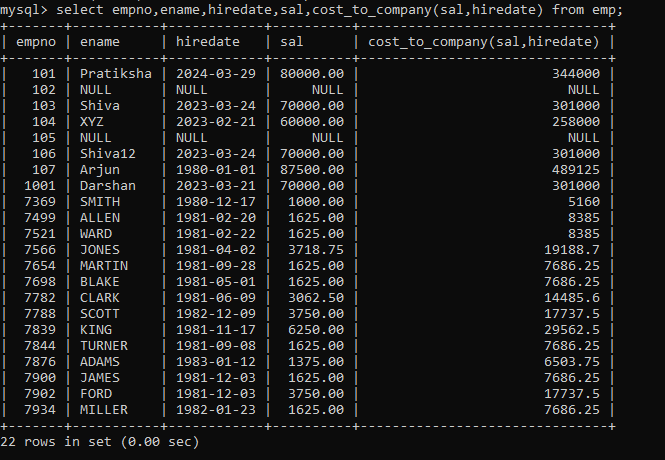
set vsal = vsal+(vsal\*0.30);

end if;

return vsal;

end //

delimiter ;



**Q2. Write trigger**

**1. Write a trigger to store the old salary details in Emp \_Back (Emp \_Back has the**

**same structure as emp table without any**

**constraint) table.**

**(note :create emp\_back table before writing trigger)**

**----- to create emp\_back table**

**create table emp\_back(**

**empno int,**

**ename varchar(20),**

**oldsal decimal(9,2),**

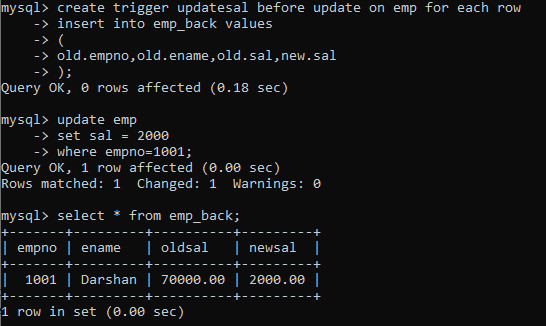
**newsal decimal(9,2)**

**)**

**(note :**

**execute procedure written in Q8 and**

**check the entries in EMP\_back table after execution of the procedure)**



**2. Write a trigger which add entry in audit table when user tries to insert or delete**

**records in employee table store empno,name,username and date on which**

**operation performed and which action is done insert or delete. in emp\_audit table.**

**create table before writing trigger.**

**create table empaudit(**

**empno int;**

**ename varchar(20),**

**username varchar(20);**

**chdate date;**

**action varchar(20)**

**);**

**create table empaudit(**

**empno int,**

**ename varchar(20),**

**job varchar(20),**

**mgr int,**

**hiredate date,**

**sal float(9,2),**

**comm float(9,2),**

**deptno int,**

**username varchar(20),**

**chdate date,**

**action varchar(20)**

**);**