**Solve the following.**

**1. write a procedure to insert record into employee table.**

**the procedure should accept empno, ename, sal, job, hiredate as input parameter**

**write insert statement inside procedure insert\_rec to add one record into table**

**create procedure insert\_rec(peno int,pnm varchar(20),psal decimal(9,2),pjob**

**varchar(20),phiredate date)**

**begin**

**insert into emp(empno,ename,sal,job,hiredate)**

**values(peno,pnm,psal,pjob,phiredate)**

**end//**

→

delimiter //

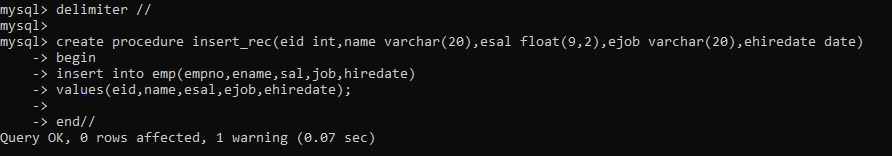
create procedure insert\_rec(eid int,name varchar(20),esal float(9,2),ejob varchar(20),ehiredate date)

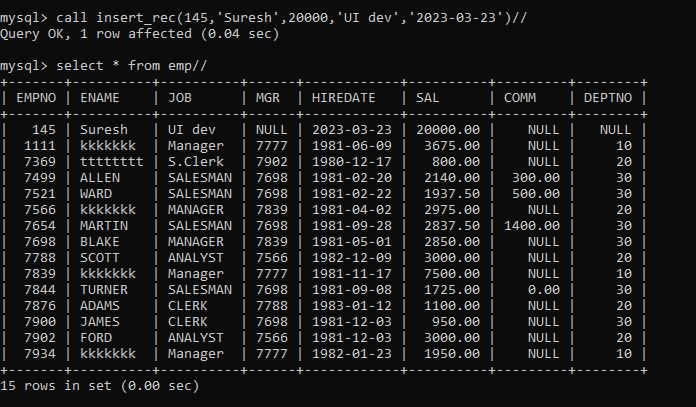
begin

insert into emp(empno,ename,sal,job,hiredate)

values(eid,name,esal,ejob,ehiredate);

end//





**2. write a procedure to delete record from employee table**

**the procedure should accept empno as input parameter.**

**write delete statement inside procedure delete\_emp to delete one record from emp Table**

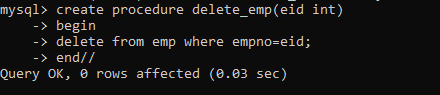
→

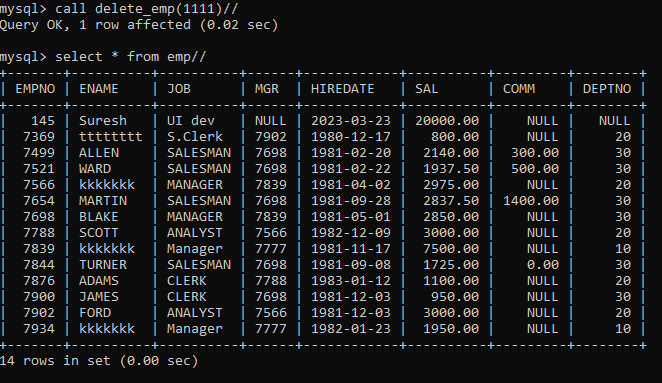
create procedure delete\_emp(eid int)

begin

delete from emp where empno=eid;

end//





**3. write a procedure to display empno,ename,deptno,dname for all employees with sal**

**> given salary. pass salary as a parameter to procedure**

→

create procedure displayEmployee(esal float(9,2))

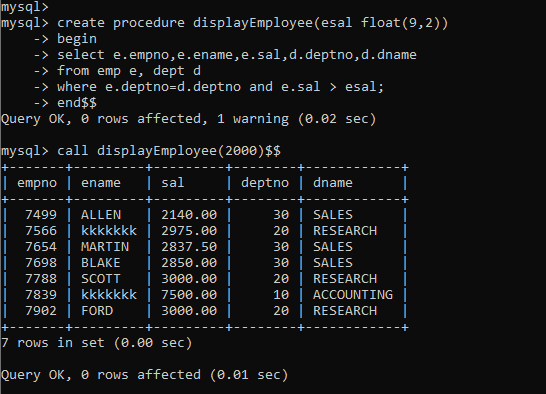
begin

select e.empno,e.ename,e.sal,d.deptno,d.dname

from emp e, dept d

where e.deptno=d.deptno and e.sal > esal;

end$$



**4. write a procedure to find min,max,avg of salary and number of employees in the**

**given deptno.**

**deptno --→ in parameter**

**min,max,avg and count ---→ out type parameter**

**execute procedure and then display values min,max,avg and count.**

→

create procedure displEmp10(ddeptno int,out minimum int,out maximum int,out average int,out cnt int)

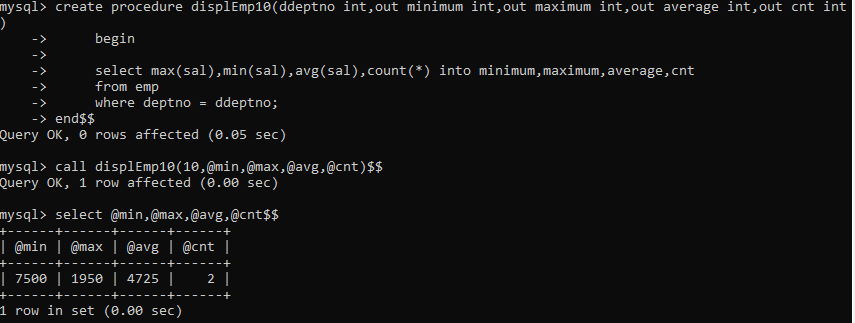
begin

select max(sal),min(sal),avg(sal),count(\*) into minimum,maximum,average,cnt

from emp

where deptno = ddeptno;

end$$



**5. write a procedure to display all pid,pname,cid,cname and salesman name(use**

**product,category and salesman table)**

delimiter //

create procedure dispProduct(in ppid int,out prodName varchar(20),out ccid int,out custName varchar(20),out salesman varchar(20))

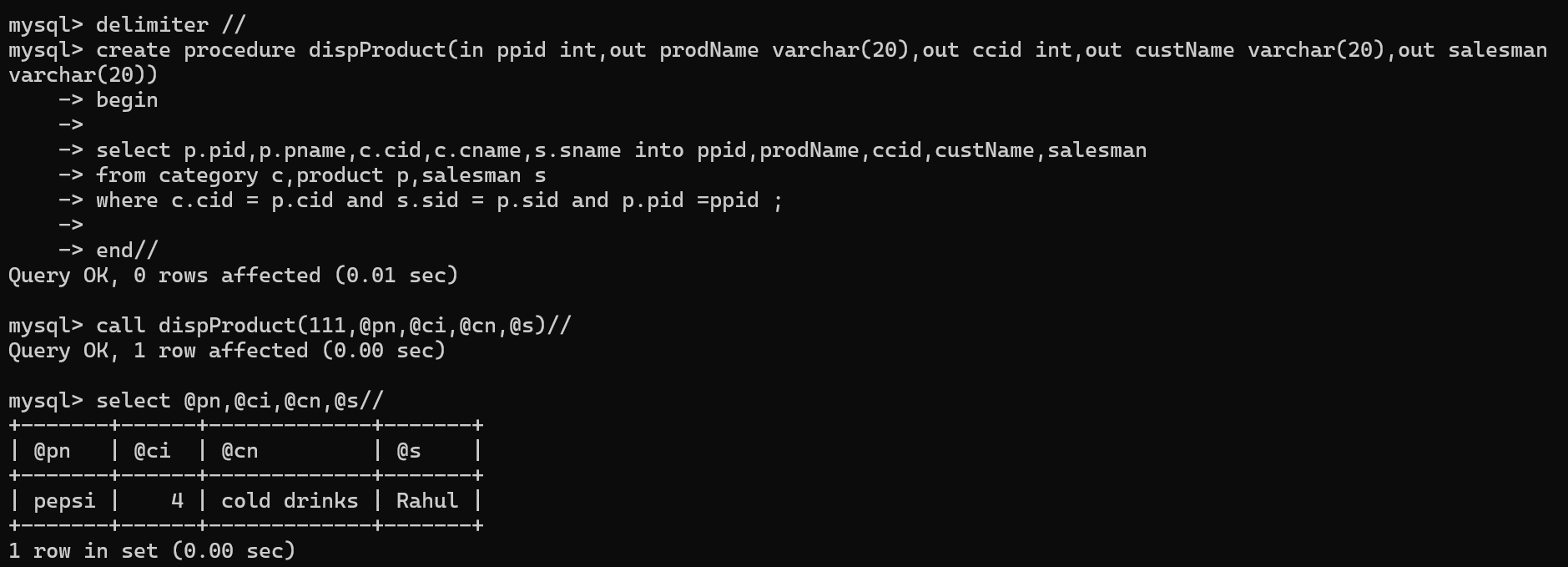
begin

select p.pid,p.pname,c.cid,c.cname,s.sname into ppid,prodName,ccid,custName,salesman

from category c,product p,salesman s

where c.cid = p.cid and s.sid = p.sid and p.pid =ppid ;

end//



**6. write a procedure to display all vehicles bought by a customer. pass customer name as**

**a parameter.(use vehicle,salesman,customer and relation table)**

→

delimiter //

create procedure dispVehicle70(in ccname varchar(20),out custName varchar(20),out vehicleName varchar(20),out salesman varchar(20))

begin

select c.cname,v.vname,s.sname into custName,vehicleName,salesman

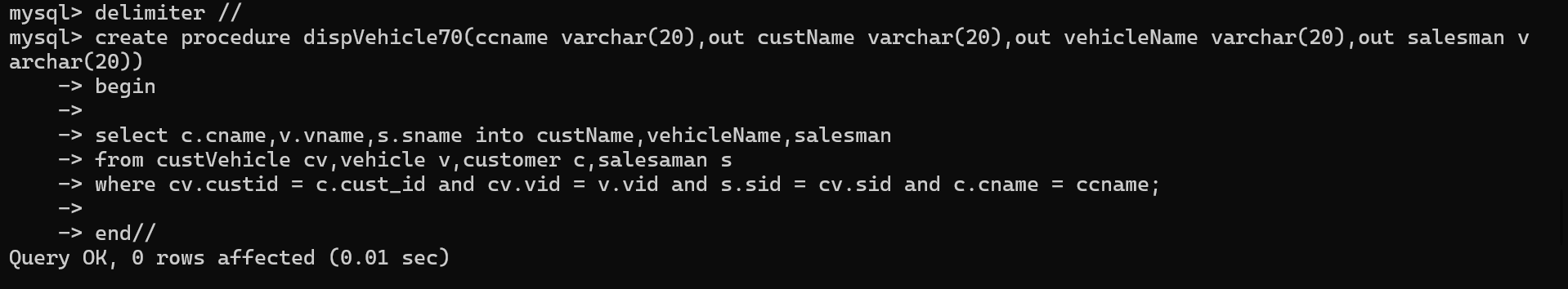
from custVehicle cv,vehicle v,customer c,salesaman s

where cv.custid = c.cust\_id and cv.vid = v.vid and s.sid = cv.sid and c.cname = ccname;

end//

call dispVehicle70('Nilima',@c,@s,@v)//

select @c,@s,@v//



**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 displayStatus()

begin

declare vfinished int default 0;

declare vempno, vdeptno int;

declare vname, vjob, status varchar(20);

declare vsal,vavg float(9,2);

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,vname, vjob, vsal, vdeptno;

if vfinished = 1 then

leave label1;

end if;

select avg(sal) into vavg from emp where deptno = vdeptno;

if vsal < vavg then

SET status = 'Lesser';

elseif vsal = vavg then

SET status = 'Equal';

else

SET status = 'Greater';

end if;

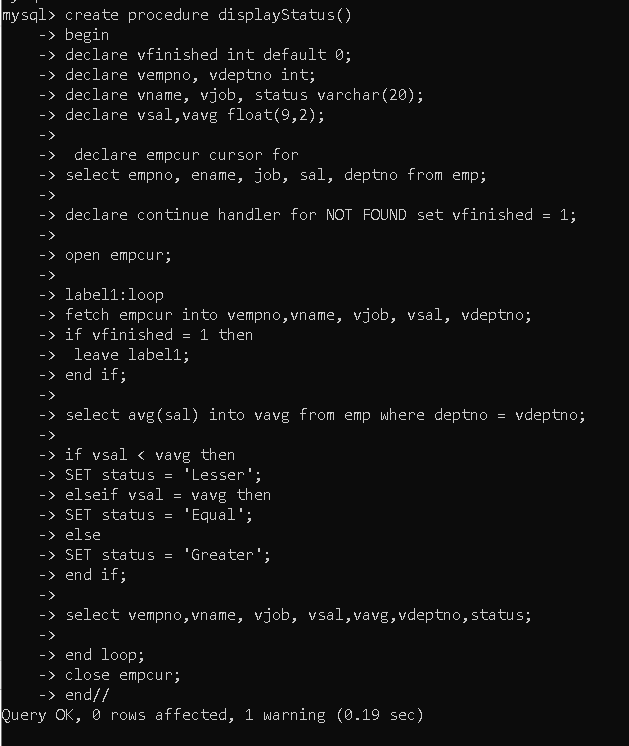
select vempno,vname, vjob, vsal,vavg,vdeptno,status;

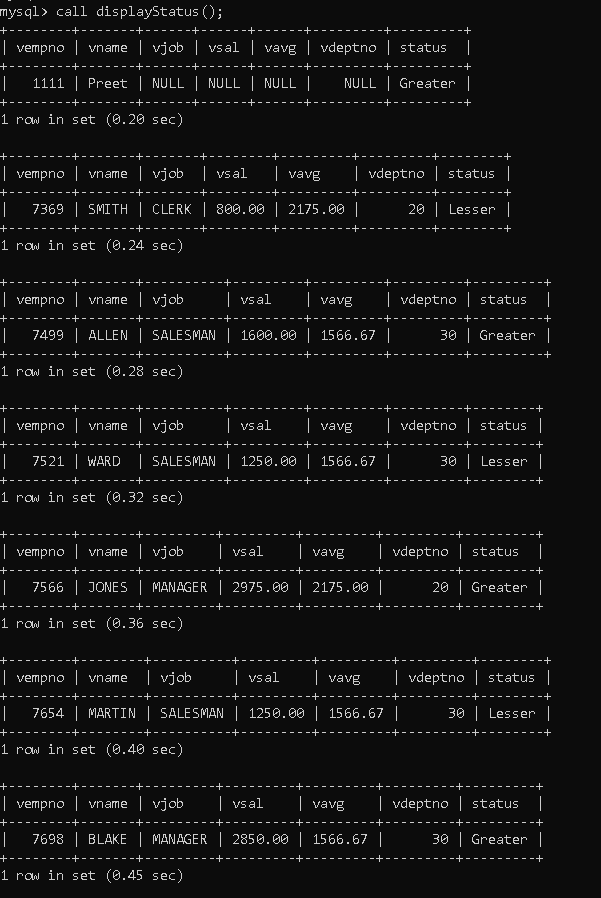
end loop;

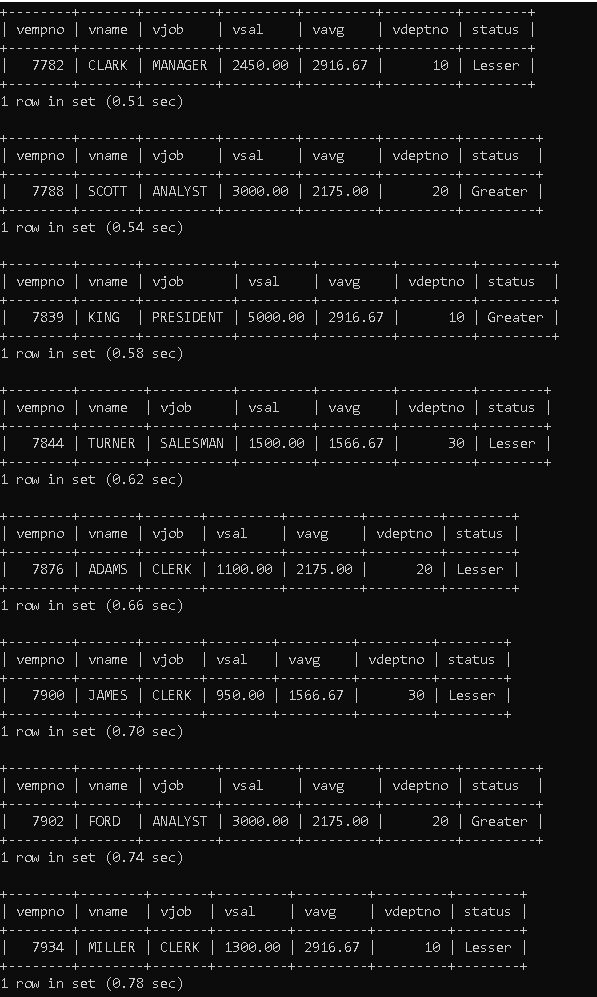
close empcur;

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**

→

create procedure updateSalary()

begin

declare vfinished int default 0;

declare vempno, vexp int;

declare vename,vjob varchar(20);

declare vhiredate date;

declare vsal,incsal float(9,2);

declare empcur cursor for

select empno, ename, job, hiredate, sal from empCpy;

declare continue handler for NOT FOUND

set vfinished = 1;

open empcur;

flag: loop

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

if vfinished = 1 then

leave flag;

end if;

set vexp = CalcExperience(vhiredate);

if vexp > 38 then

set incsal = vsal \* 1.25;

update empCpy

set sal = incsal

where empno = vempno;

elseif vexp > 35 then

set incsal = vsal \* 1.2;

update empCpy

set sal = incsal

where empno = vempno;

else

update empCpy

set sal = vsal

where empno = vempno;

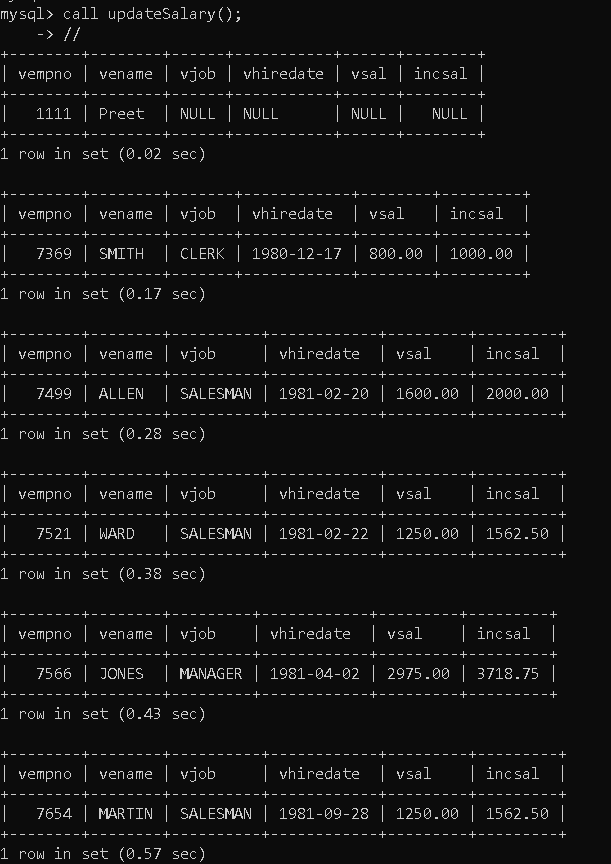
end if;

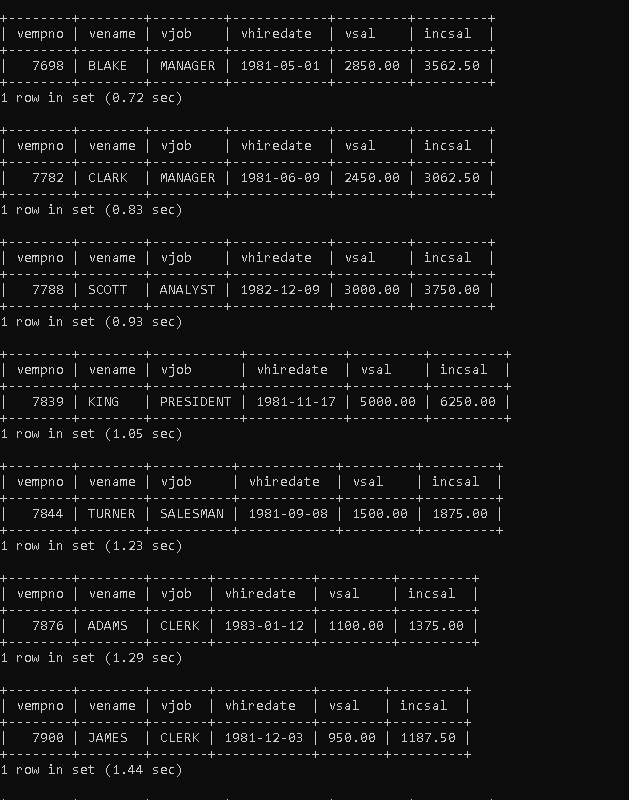
select vempno,vename, vjob, vhiredate, vsal,incsal;

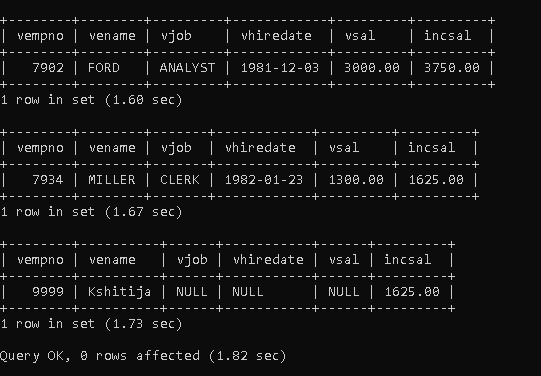
end loop;

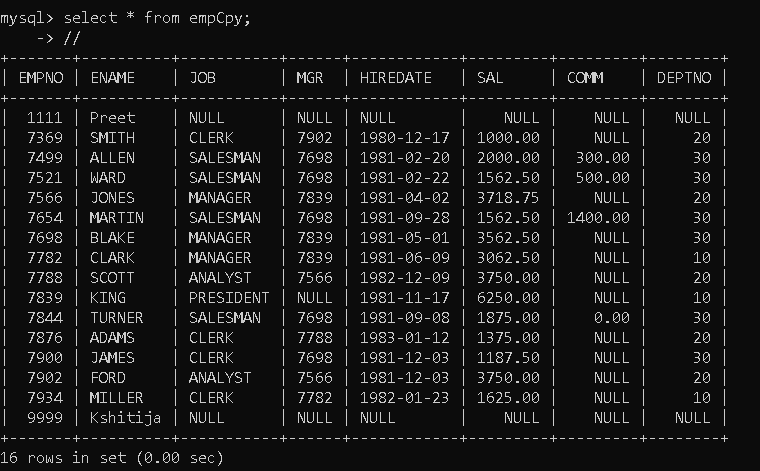
end//











**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));**

**→**

mysql>

create table emp\_allowance(

empno int,

ename varchar(20),

hiredate date,

experience int,

allowance decimal(9,2)

);

mysql>

create function calcExp9(ehiredate date) returns int

begin

declare exp int;

SET exp = floor(datediff(curdate(),ehiredate)/365);

return exp;

end//

mysql>

create procedure addAllowance()

begin

declare vfinished int default 0;

declare vempno, vexp int;

declare vename varchar(20);

declare vhiredate date;

declare vallowance decimal(9,2);

declare empcur cursor for

select empno,ename,hiredate from empCpy;

declare continue handler for NOT FOUND

set vfinished = 1;

open empcur;

flag:loop

fetch empcur into vempno, vename, vhiredate;

if vfinished = 1 then

leave flag;

end if;

set vexp = calcExp9(vhiredate);

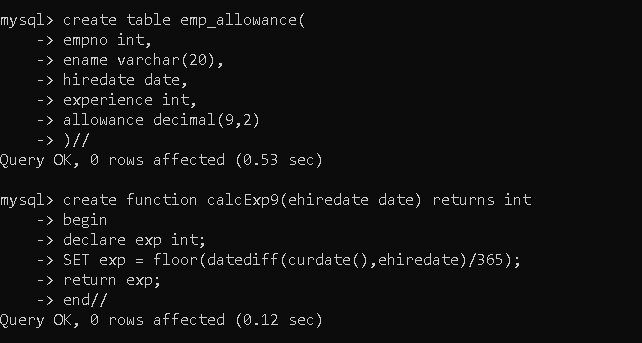
set vallowance = vexp \* 3000;

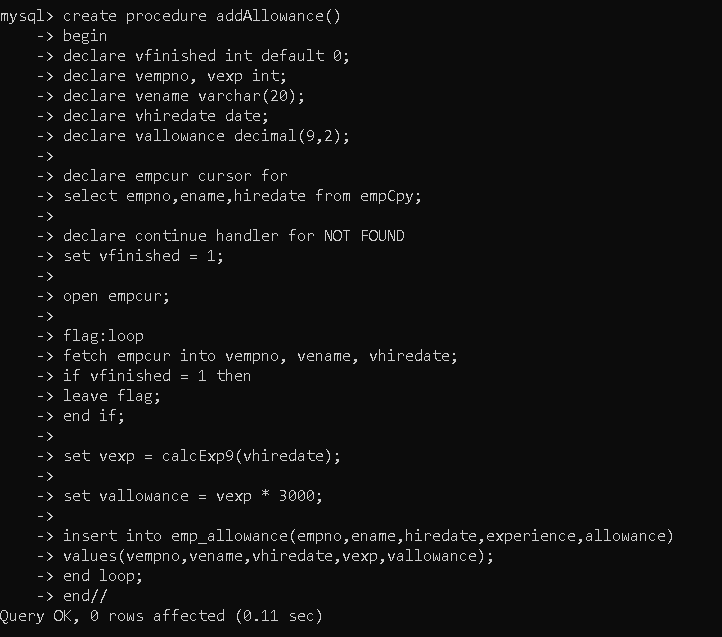
insert into emp\_allowance(empno,ename,hiredate,experience,allowance)

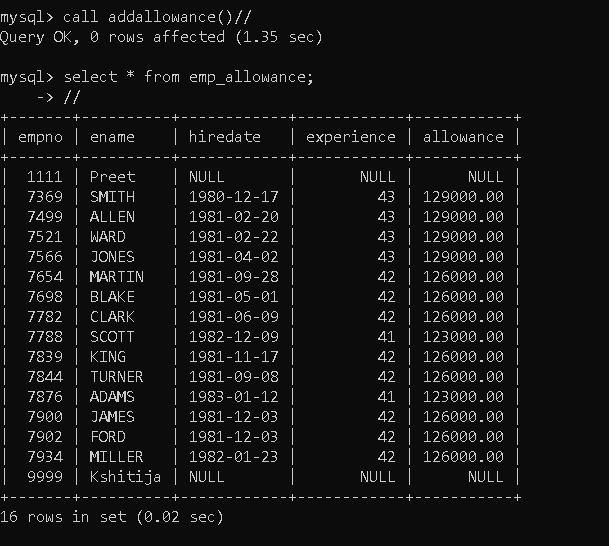
values(vempno,vename,vhiredate,vexp,vallowance);

end loop;

end//

****

****

****

**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**

**→**

create function calc\_CTC(sal decimal(9,2),hiredate date) returns decimal

begin

declare ctc,incsal decimal(9,2);

declare exp int;

set exp = calcExperience(hiredate);

if exp>4 then

set incsal = sal\*1.3 + 0.15\*sal + 0.20\*sal + 0.08\*sal;

elseif exp>=2 and exp<4 then

set incsal = sal\*1.2 + 0.15\*sal + 0.20\*sal + 0.08\*sal;

elseif exp>=1 and exp<2 then

set incsal = sal\*1.1 + 0.15\*sal + 0.20\*sal + 0.08\*sal;

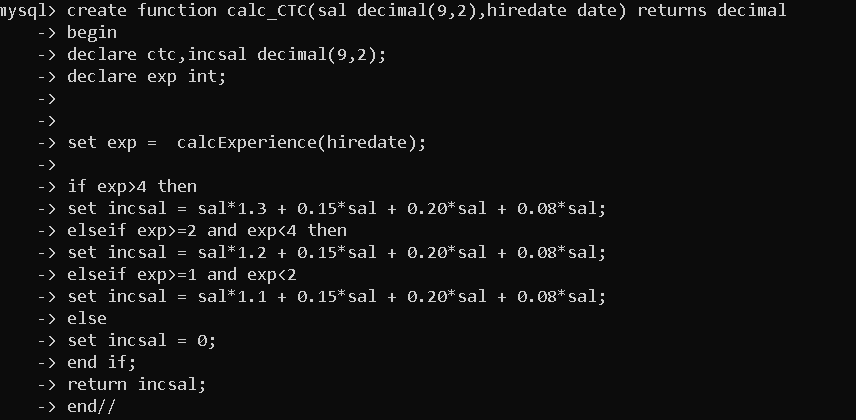
else

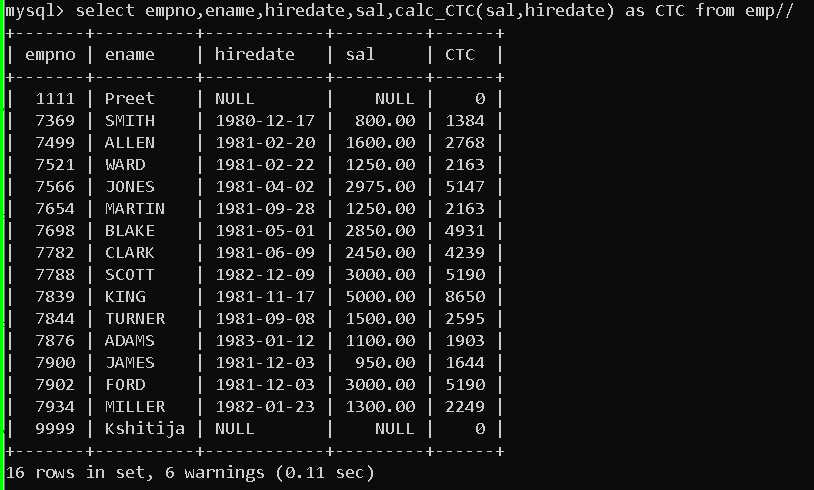
set incsal = 0;

end if;

return incsal;

end//

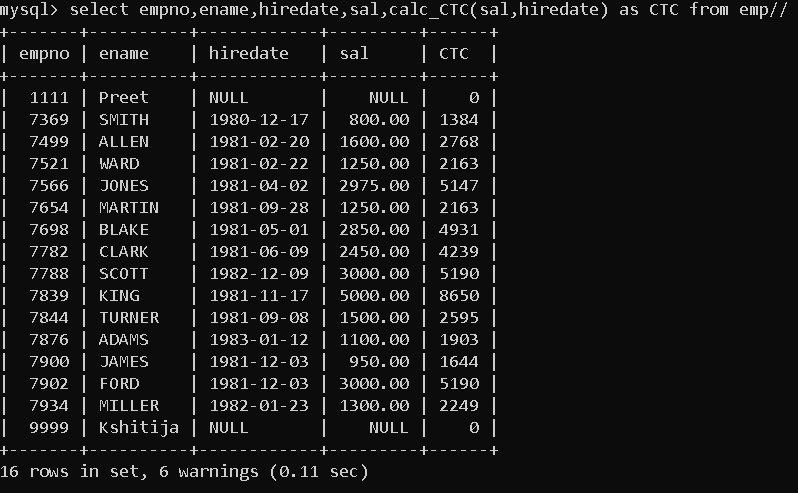
****

****

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

**use function written in question 10)**

**→**

****

**Q2. Write trigger**

**1. Write a tigger 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)**

**→**

mysql>

create table emp\_back(

empno int,

ename varchar(20),

oldsal decimal(8,2),

newsal decimal(8,2),

action varchar(10)

);

mysql>

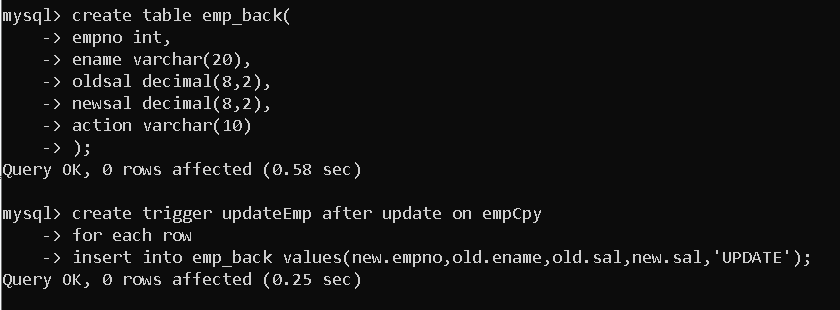
create trigger updateEmp after update on empCpy

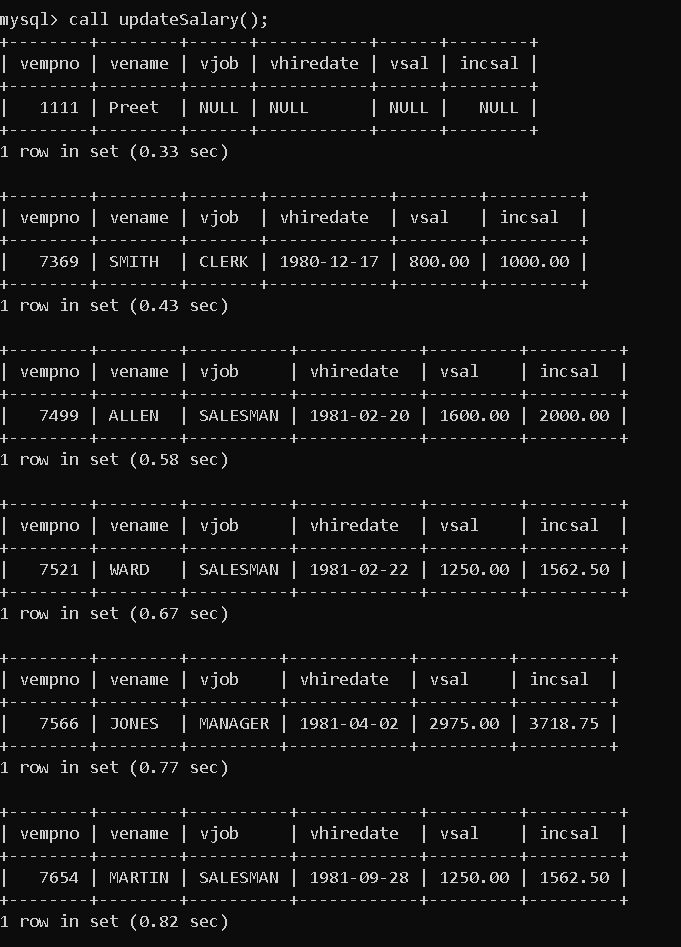
for each row

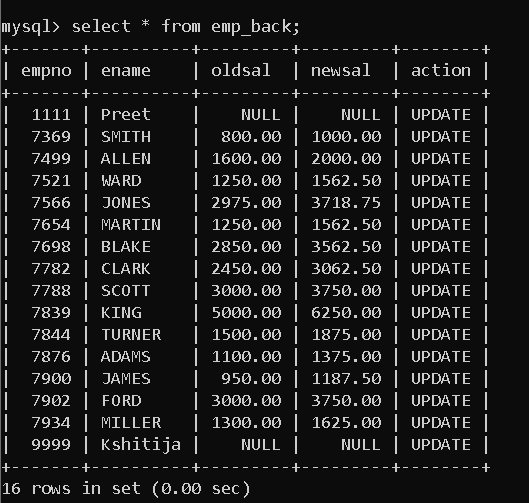
insert into emp\_back values(new.empno,old.ename,old.sal,new.sal,'UPDATE');

mysql>

call updateSalary();







**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)**

**);**

→

mysql> create table emp\_audit(

empno int,

oldename varchar(20),

newename varchar(20),

user varchar(20),

change\_date date,

action varchar(10)

);

Query OK, 0 rows affected (3.20 sec)

Insert Trigger :

mysql> create trigger insertEmp after insert on emp

for each row

insert into emp\_audit values

(new.empno,null,new.ename,user(),now(),'I NSERT');

Query OK, 0 rows affected (0.71 sec)

mysql> insert into emp(empno,ename) values(9999,'Kshitija');

Query OK, 1 row affected (0.41 sec)

mysql> insert into emp(empno,ename,job,mgr,hiredate,sal,comm,deptno) values

(8888,'Leena','Manager',7092,'2023-10-16',80000,800,20);

Query OK, 1 row affected (0.21 sec)



Can write multiple insert trigger with specific column :

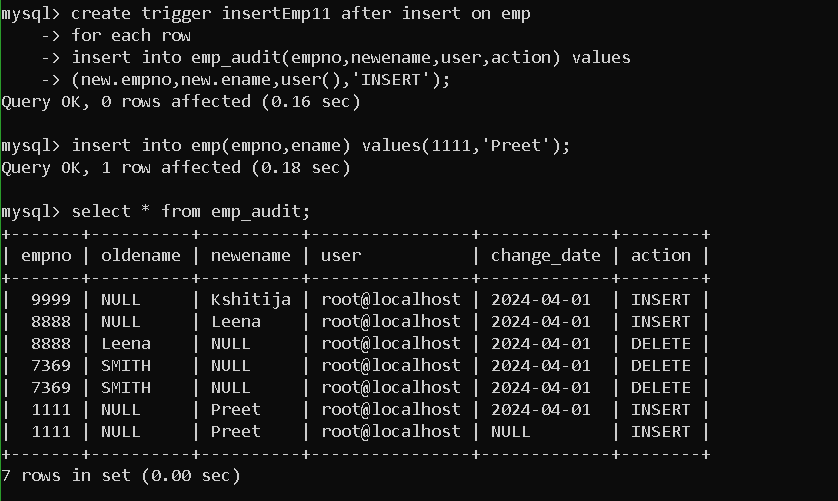
mysql> create trigger insertEmp11 after insert on emp

for each row

insert into emp\_audit(empno,newename,user,action) values

(new.empno,new.ename,user(),'I NSERT');

mysql> insert into emp(empno,ename) values(1111,'Preet');



Delete Trigger :

mysql> create trigger deleteEmp after delete on emp

for each row

insert into emp\_audit values

(old.empno,old.ename,null,user(),now(),'DELETE');

mysql>create trigger deleteEmp11 after delete on emp

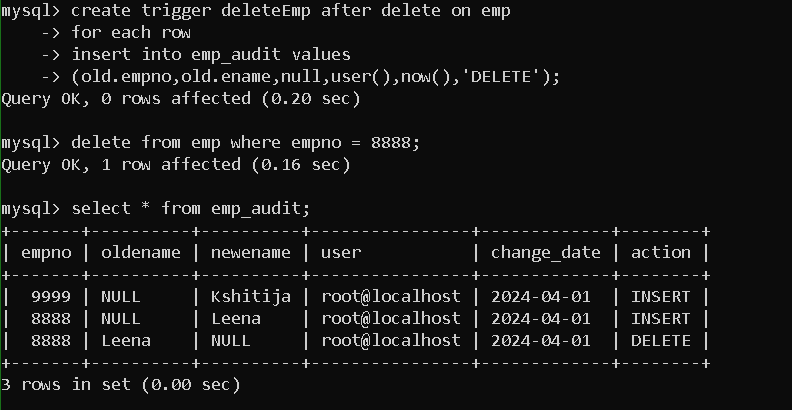
for each row

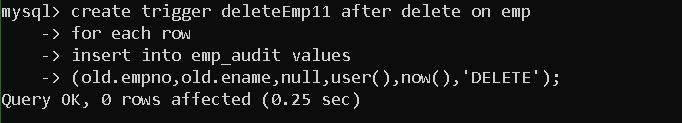
insert into emp\_audit values

(old.empno,old.ename,null,user(),now(),'DELETE');

( Can create multiple triggers, it will execute both triggers on performing delete operation)

mysql> delete from emp where empno = 8888;





**3. Create table vehicle\_history. Write a trigger to store old vehicleprice and new vehicle**

**price in history table before you update price in vehicle table**

**(note: use vehicle table).**

**create table vehicle\_history(**

**vno int,**

**vname varchar(20),**

**oldprice decimal(9,2),**

**newprice decimal(9,2),**

**chdate date,**

**username varchar(20)**

→

create table vehicle\_history(

vno int,

vname varchar(20),

oldprice decimal(9,2),

newprice decimal(9,2),

changeTime datetime,

user varchar(20),

action varchar(20));

create trigger updateVehicle after update on vehicle

for each row

insert into vehicle\_history values

(old.vid,old.vname,old.price,new.price,now(),user(),'UPDATE');

update vehicle

set price = 90000 where vid=1;

