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 //**

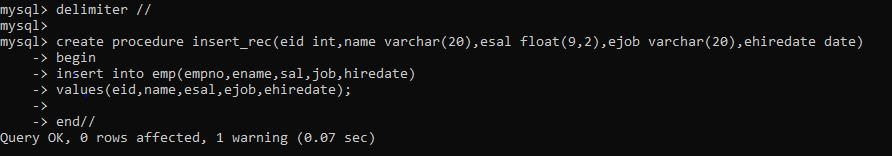
**create procedure insert\_rec(eid int,name varchar(20),esal float(9,2),ejob varchar(20),ehiredate date)**

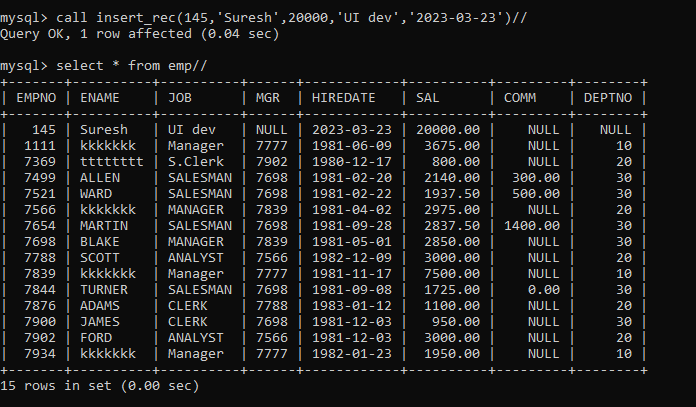
**begin**

**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

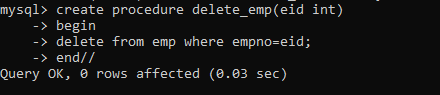
→

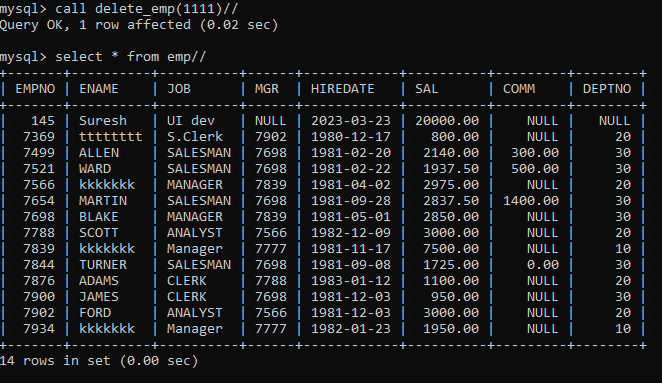
create procedure delete\_emp(eid int)

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))

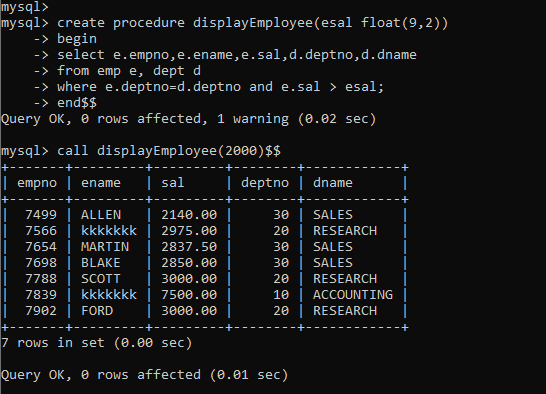
begin

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)

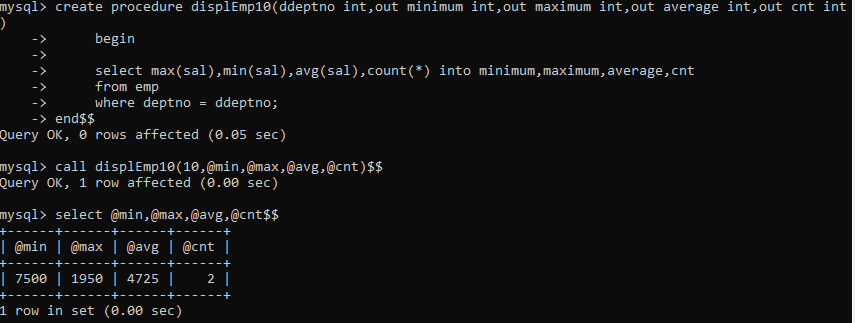
begin

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

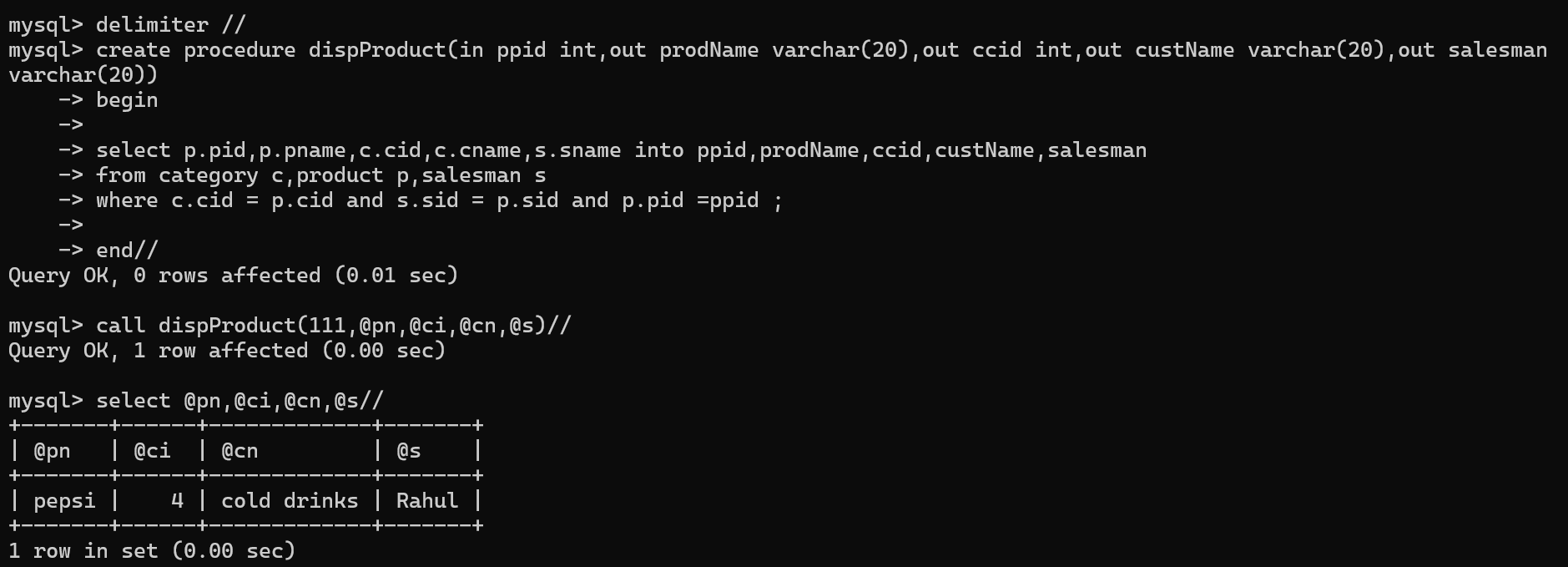
select p.pid,p.pname,c.cid,c.cname,s.sname into ppid,prodName,ccid,custName,salesman

from category c,product p,salesman s

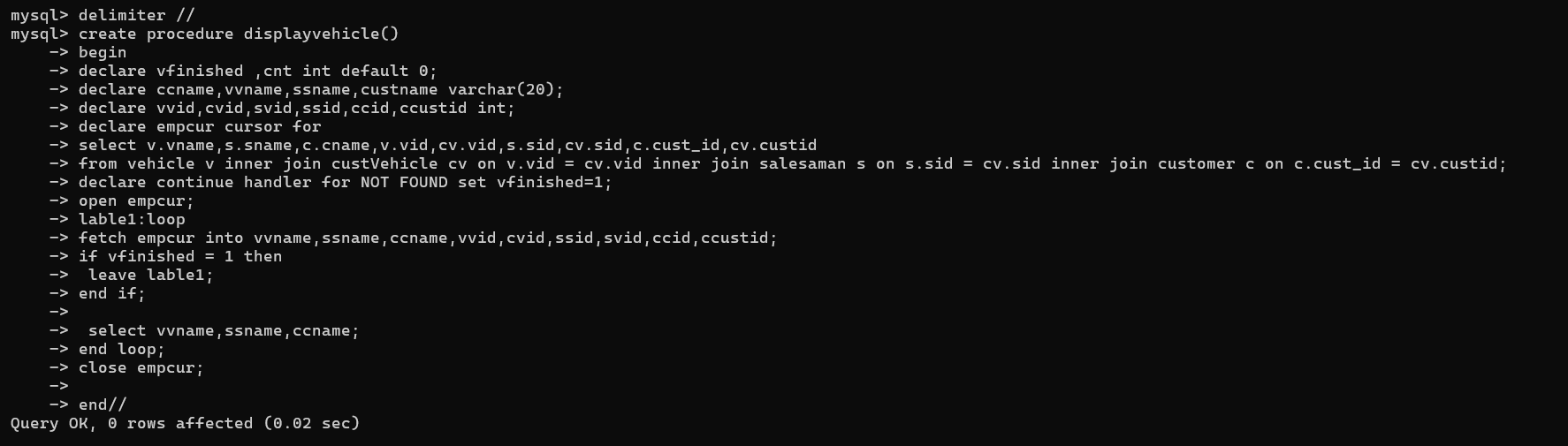
where c.cid = p.cid and s.sid = p.sid and p.pid =ppid ;

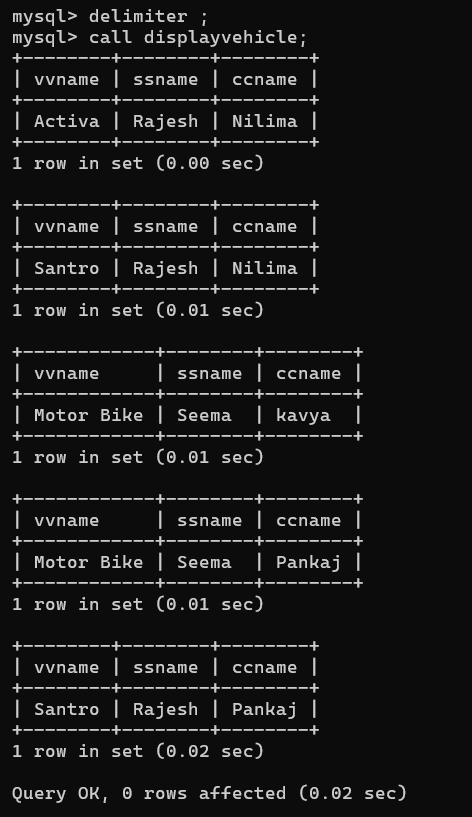
end//

Using procedure→



Using cursor→





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

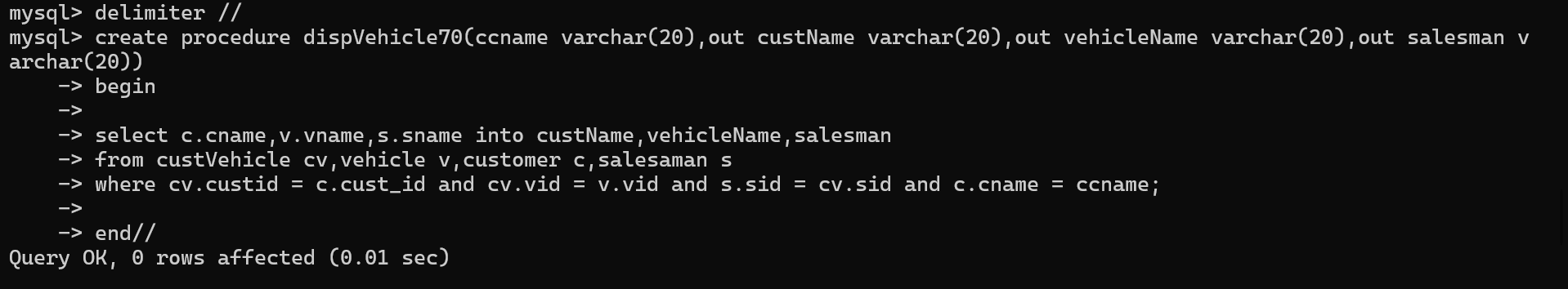
from custVehicle cv,vehicle v,customer c,salesaman s

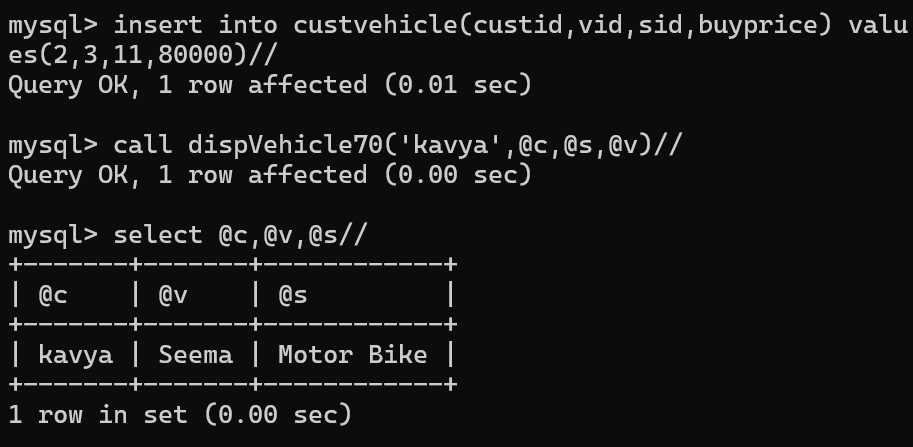
where cv.custid = c.cust\_id and cv.vid = v.vid and s.sid = cv.sid and c.cname = ccname;

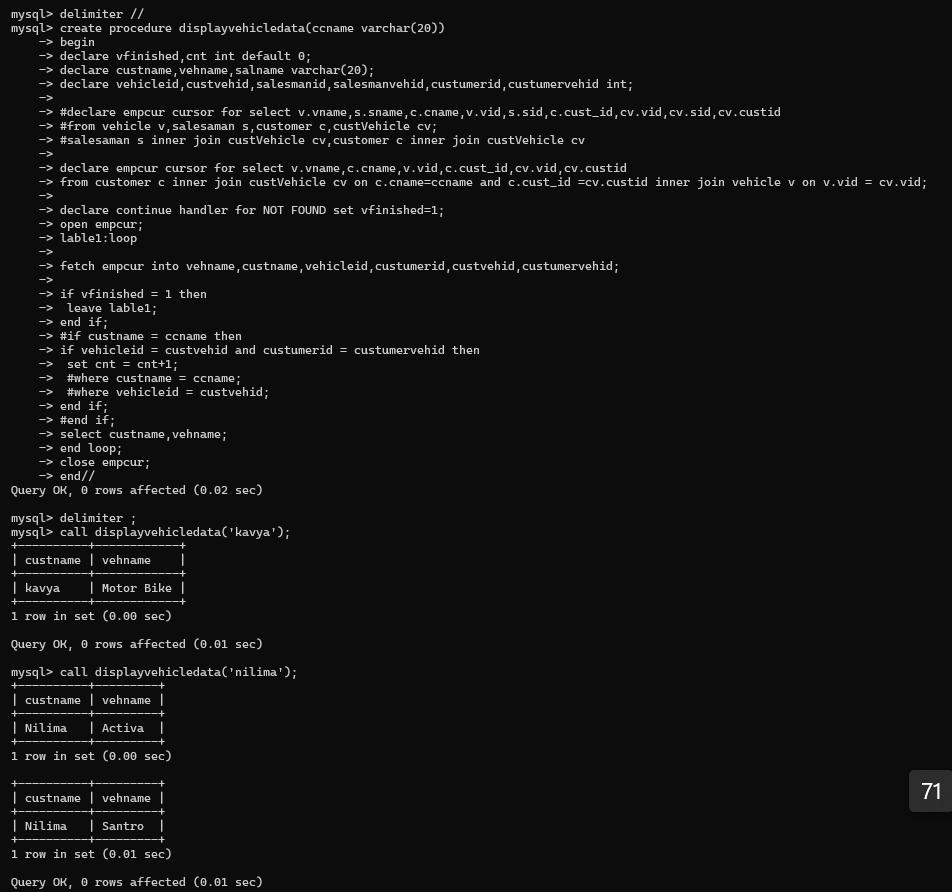
end//

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

select @c,@s,@v//







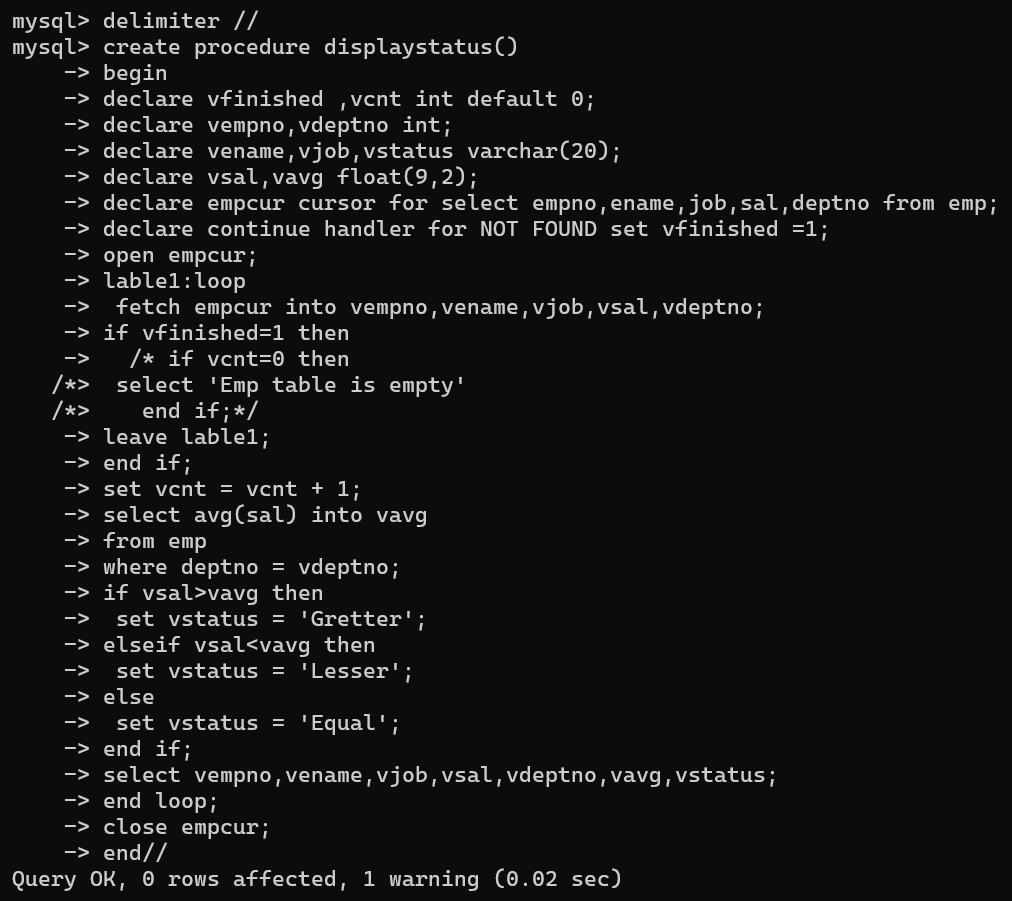
7. Write a procedure that displays the following information of all emp

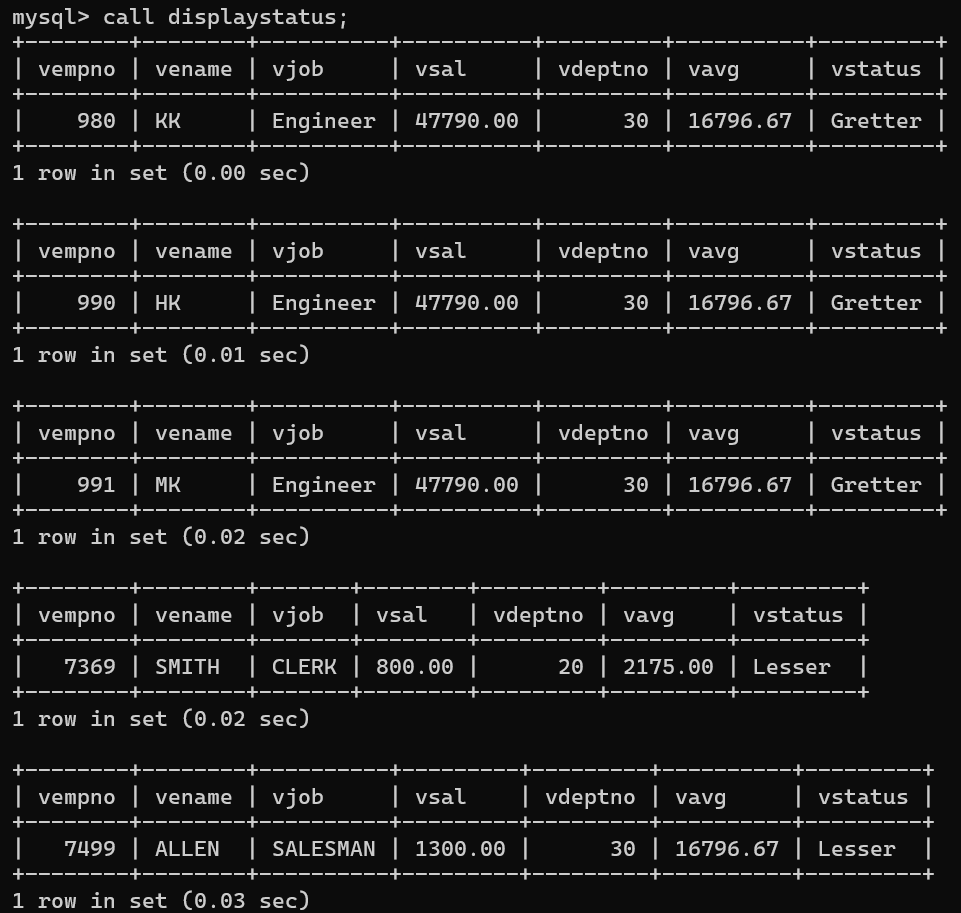
Empno,Name,job,Salary,Status,deptno

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.





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 updatesalaryemp()

begin

declare vexp int default 0;

declare vfinished int default 0;

declare vempno int;

declare vename, vjob varchar(20);

declare vsal,incsal float(9,2);

declare vhiredate date;

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

declare continue handler for NOT FOUND set vfinished=1;

open empcur;

lable1:loop

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

if vfinished =1 then

leave lable1;

end if;

select calcexp(vhiredate) into vexp from emp

where empno = vempno;

#set vexp = calcexperience(vhiredate);

if vexp>35 and vexp<=38 then

#set vexp = calcexp(vhiredate);

set incsal = vsal \* 1.2;

update emp

set sal = incsal

where empno = vempno;

elseif vexp>38 then

set incsal = vsal \* 1.25;

update emp

set sal = incsal

where empno = vempno;

else

set incsal = vsal;

update emp

set sal = vsal

where empno = vempno;

end if;

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

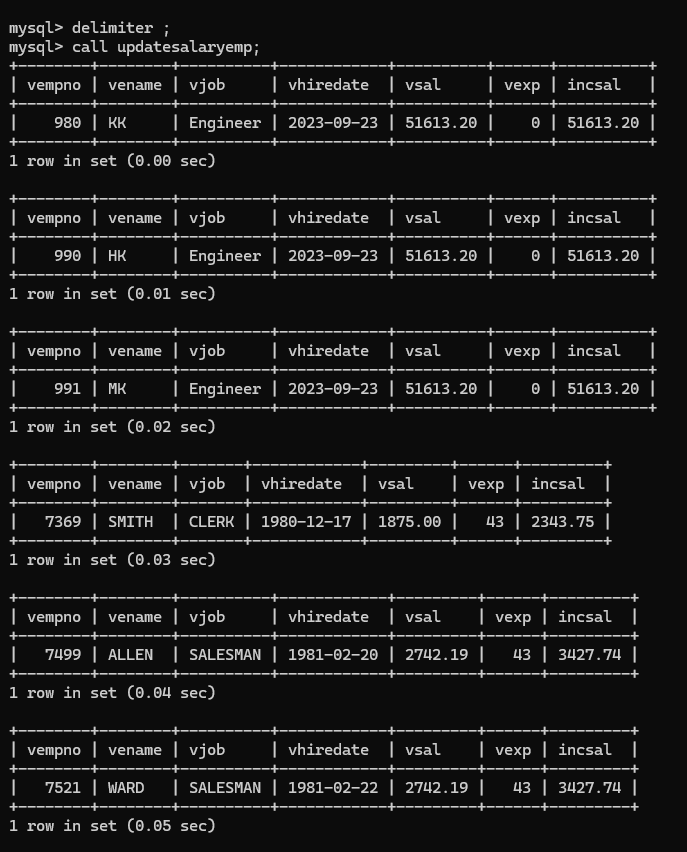
end loop;

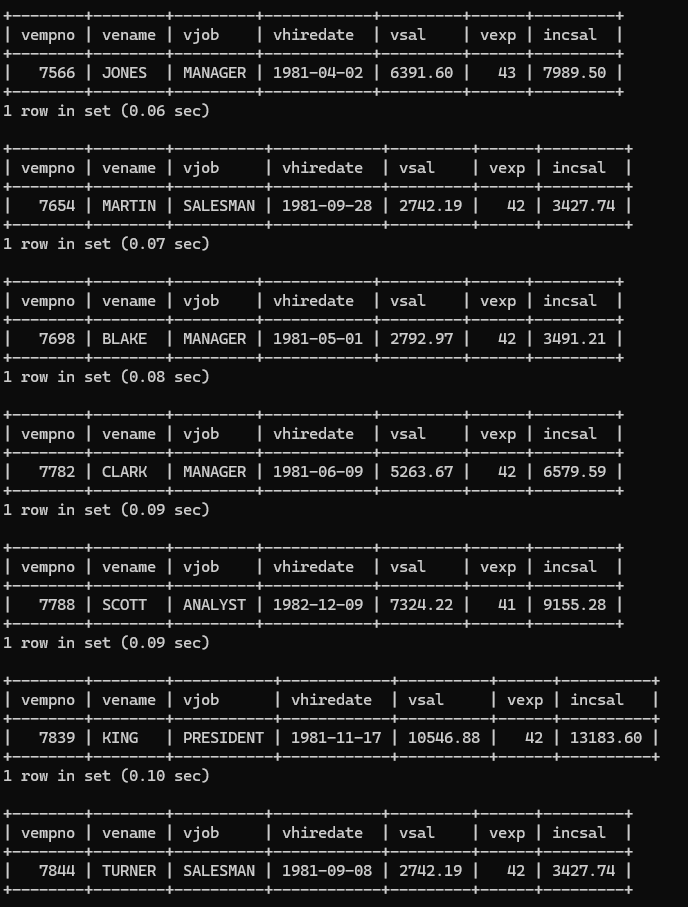
close empcur;

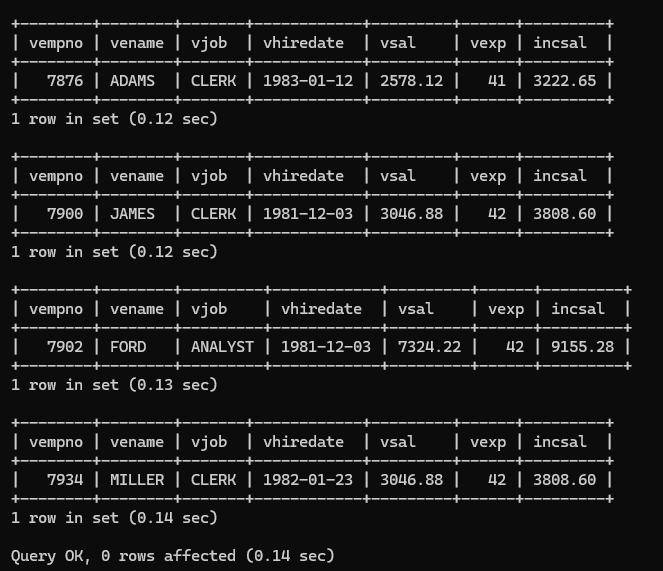
end//

delimiter ;









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(

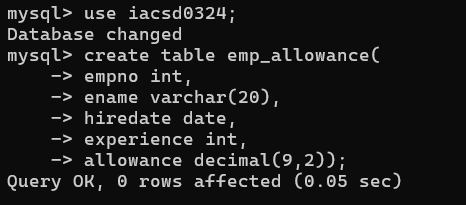
empno int,

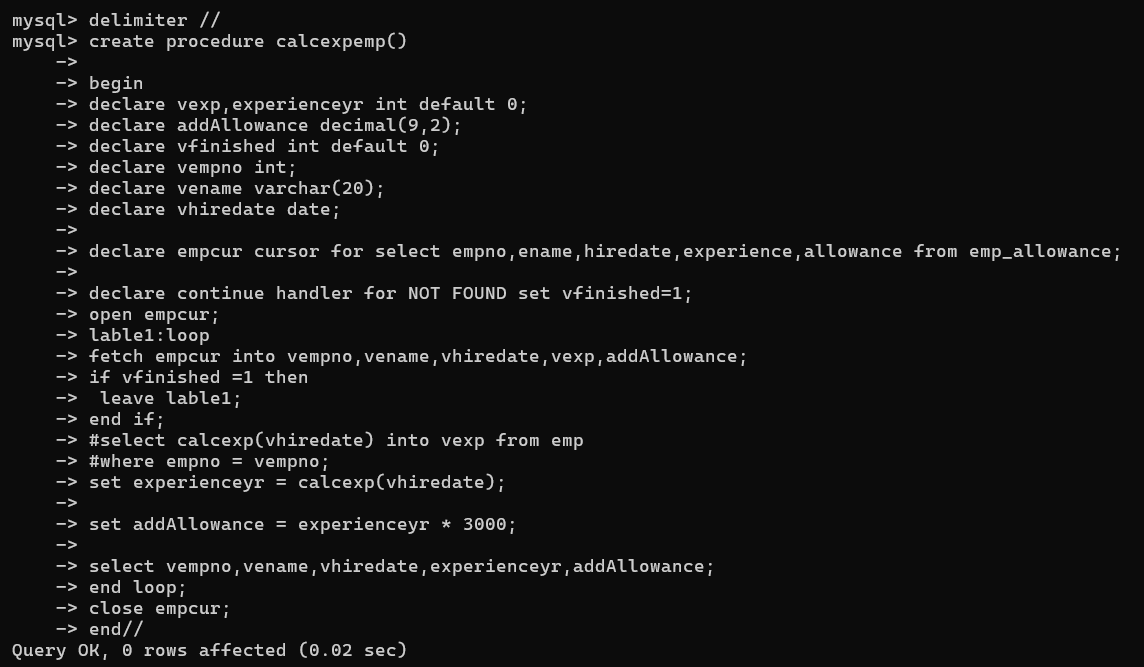
ename varchar(20),

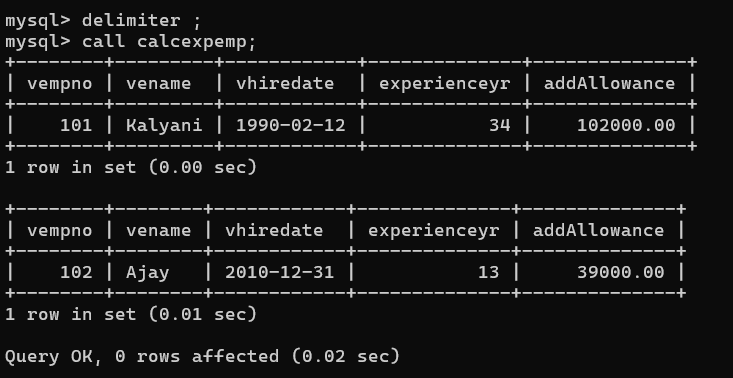
hiredate date,

experience int,

allowance decimal(9,2));







delimiter //

create procedure calcexpemp()

begin

declare vexp,experienceyr int default 0;

declare vsal,vcomm,addAllowance decimal(9,2);

declare vfinished int default 0;

declare vempno,vdeptno,vmgr int;

declare vename,vjob varchar(20);

declare vhiredate date;

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

declare continue handler for NOT FOUND set vfinished=1;

open empcur;

lable1:loop

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

if vfinished =1 then

leave lable1;

end if;

#select calcexp(vhiredate) into vexp from emp

#where empno = vempno;

set experienceyr = calcexp(vhiredate);

set addAllowance = experienceyr \* 3000;

select vempno,vename,vhiredate,experienceyr,addAllowance;

#INSERT INTO emp\_allowance(empno, ename, hiredate, experience, allowance)

# VALUES(vempno, vename, vhiredate, experienceyr, addAllowance);

insert into emp\_allowance values(vempno,vename,vjob,vmgr,vhiredate,vsal,vcomm,vdeptno,experienceyr,addAllowance,user(),now(),'insert');

end loop;

close empcur;

end//

delimiter ;

create table emp\_allowance(

empno int,

name varchar(20),

ejob varchar(20),

emgr int,

ehiredate date,

esal float(9,2),

ecomm int,

edeptno int,

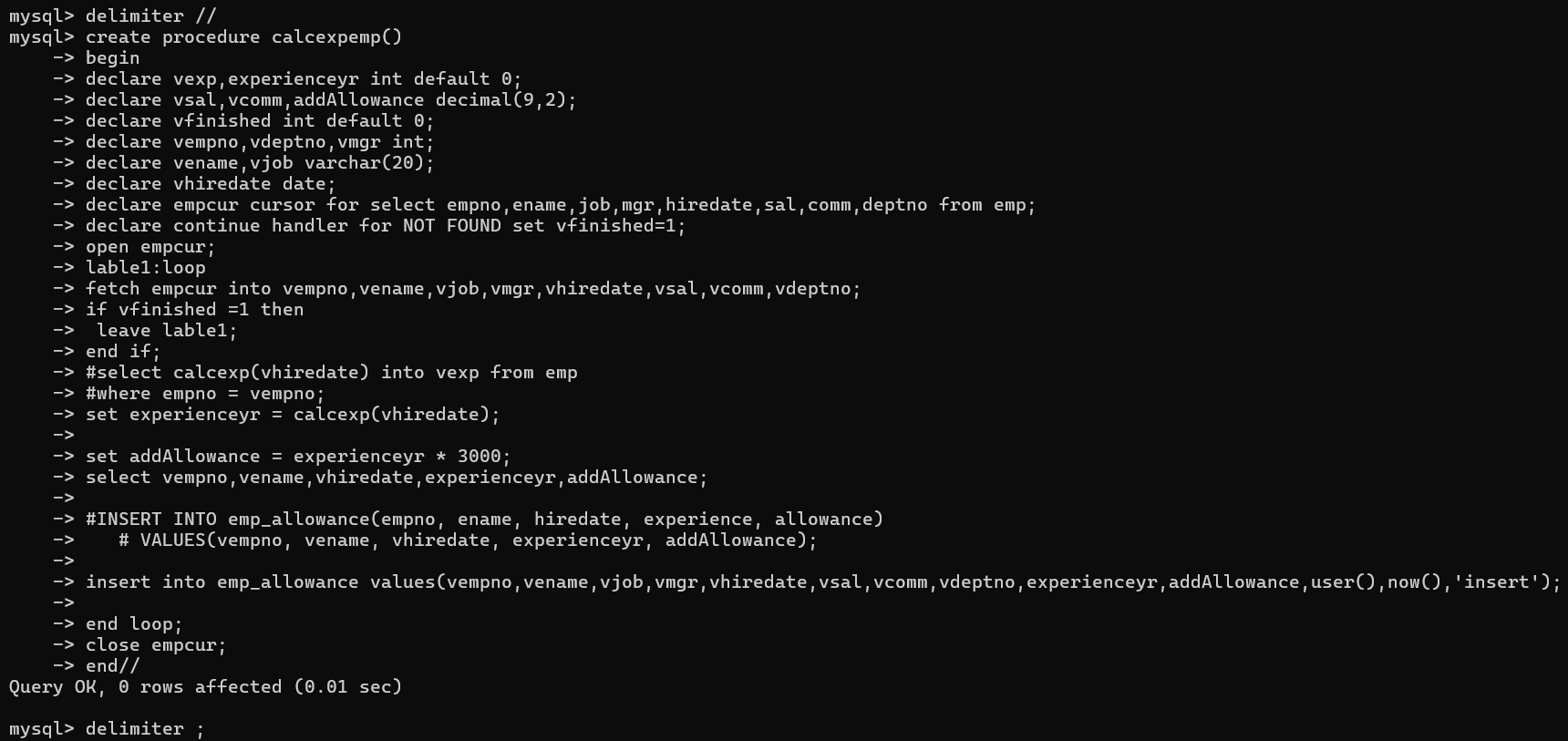
expyr int,

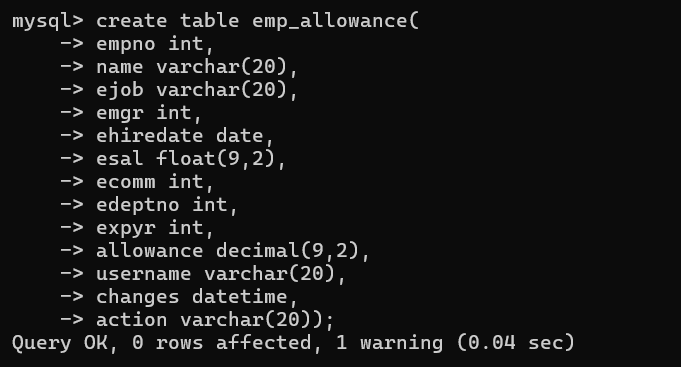
allowance decimal(9,2),

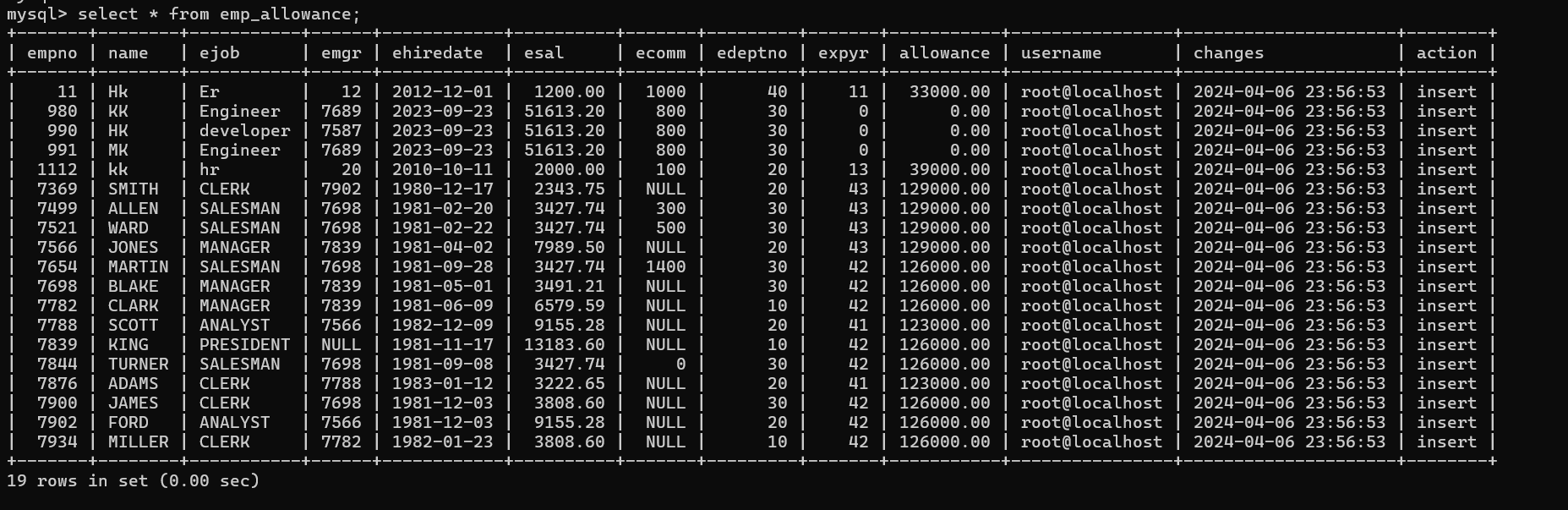
username varchar(20),

changes datetime,

action varchar(20));







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.

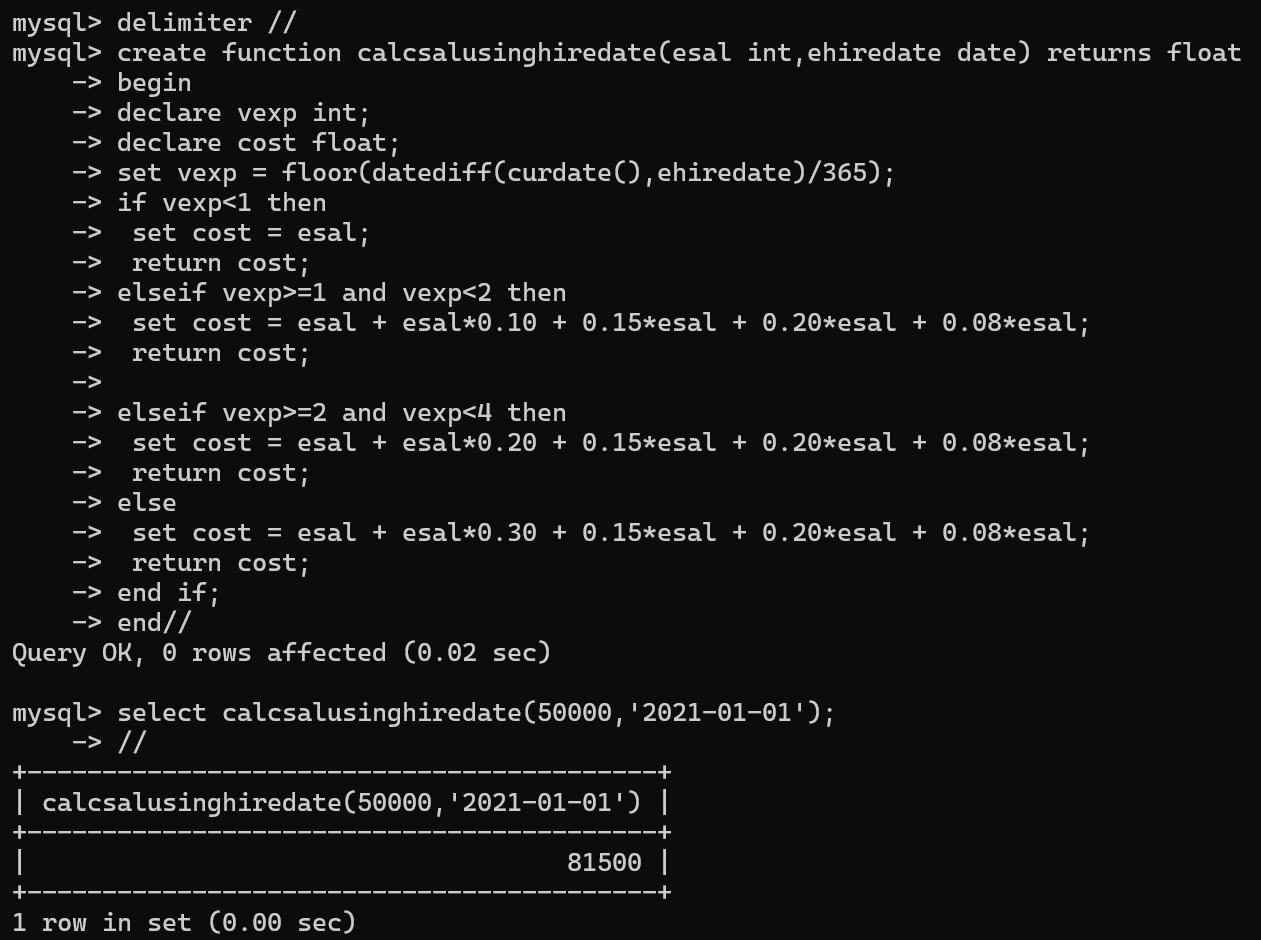
Special Allowance will be decided based on the service in the company.

< 1 Year Nil

>=1 Year< 2 Year 10% of Salary

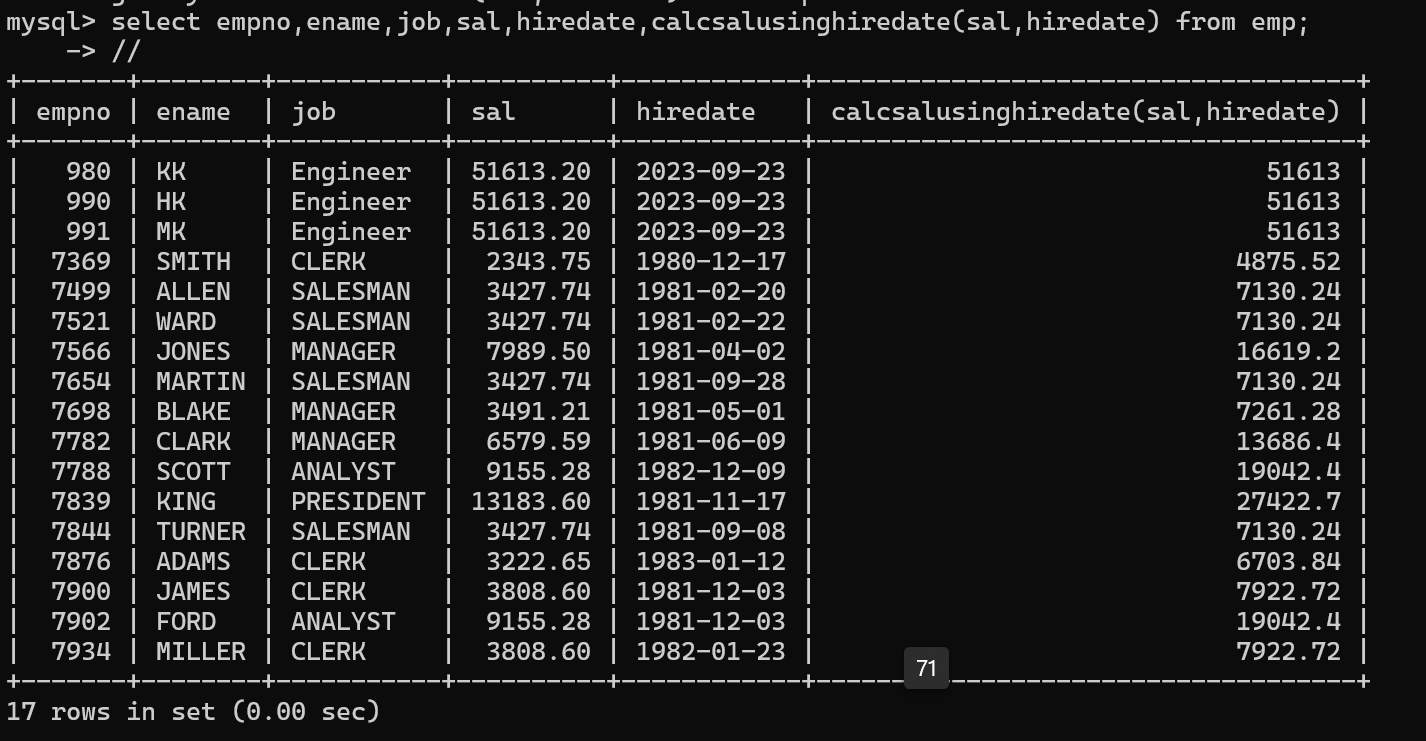
>=2 Year< 4 Year 20% of Salary

>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)



**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)

create table emp\_audit(

empno int,

ename varchar(20),

oldsal decimal(9,2),

newsal decimal(9,2),

uname varchar(20),

changes datetime,

action varchar(20));

delimiter //

create procedure updateemp()

begin

declare vfinished int default 0;

declare vempno,vexp int;

declare vename varchar(20);

declare vsal,nsal,incsal decimal(9,2);

declare vhiredate date default '1990-01-01';

declare empcur cursor for select empno,ename,oldsal,newsal from emp\_back;

declare continue handler for NOT FOUND set vfinished=1;

open empcur;

lable1:loop

fetch empcur into vempno,vename,vsal,nsal;

if vfinished =1 then

leave lable1;

end if;

#select calcexp(vhiredate) into vexp from emp

#where empno = vempno;

set vexp = calcexp(vhiredate);

if vexp>35 and vexp<=38 then

set incsal = nsal \* 1.2;

update emp\_back

set newsal = incsal

where empno = vempno;

elseif vexp>38 then

set incsal = nsal \* 1.25;

update emp\_back

set newsal = incsal

where empno = vempno;

else

set incsal = nsal;

update emp\_back

set newsal = nsal

where empno = vempno;

end if;

select vempno,vename,vsal,nsal,vexp,incsal;

end loop;

close empcur;

end//

delimiter ;

create trigger insertempback after insert on emp\_back

for each row

insert into emp\_audit values(

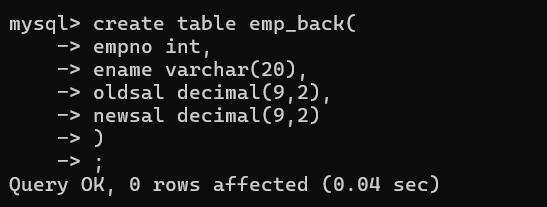
new.empno,new.ename,null,new.newsal,user(),now(),'insert');

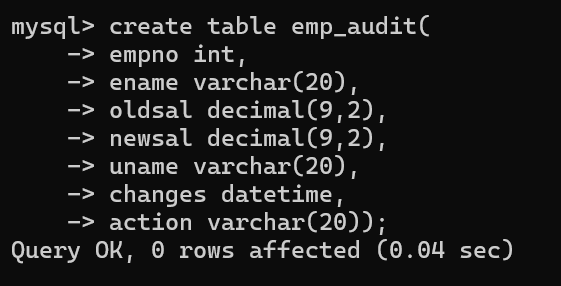
create trigger updateempback after update on emp\_back

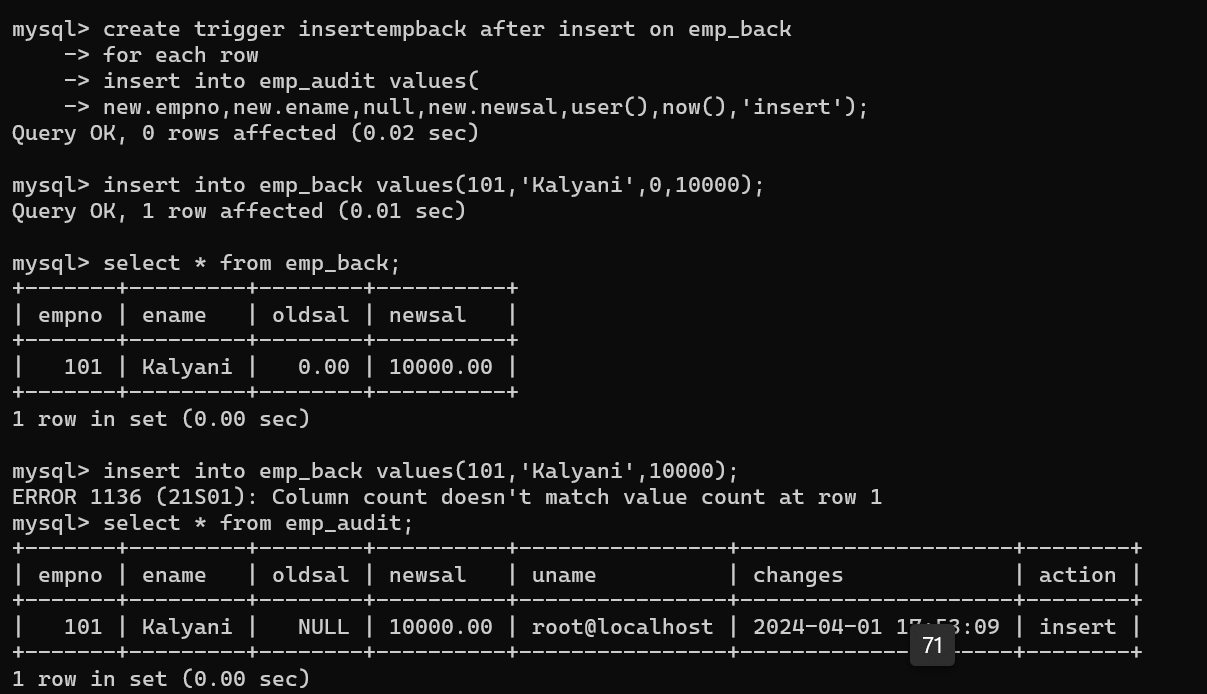
for each row

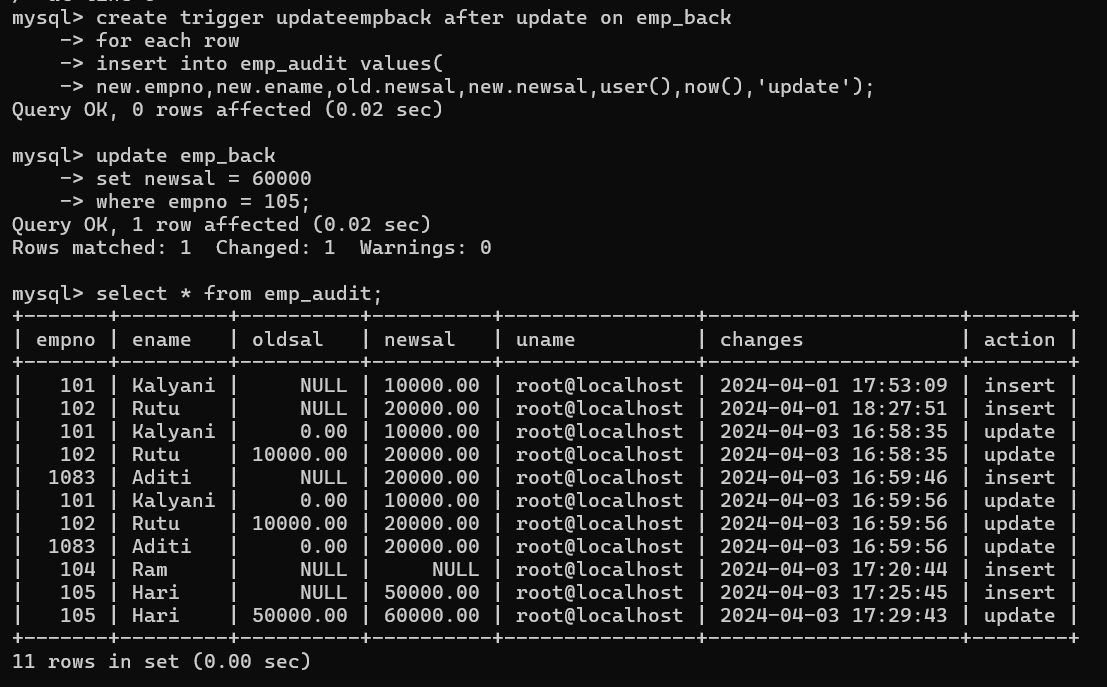
insert into emp\_audit values(

new.empno,new.ename,old.newsal,new.newsal,user(),now(),'update');

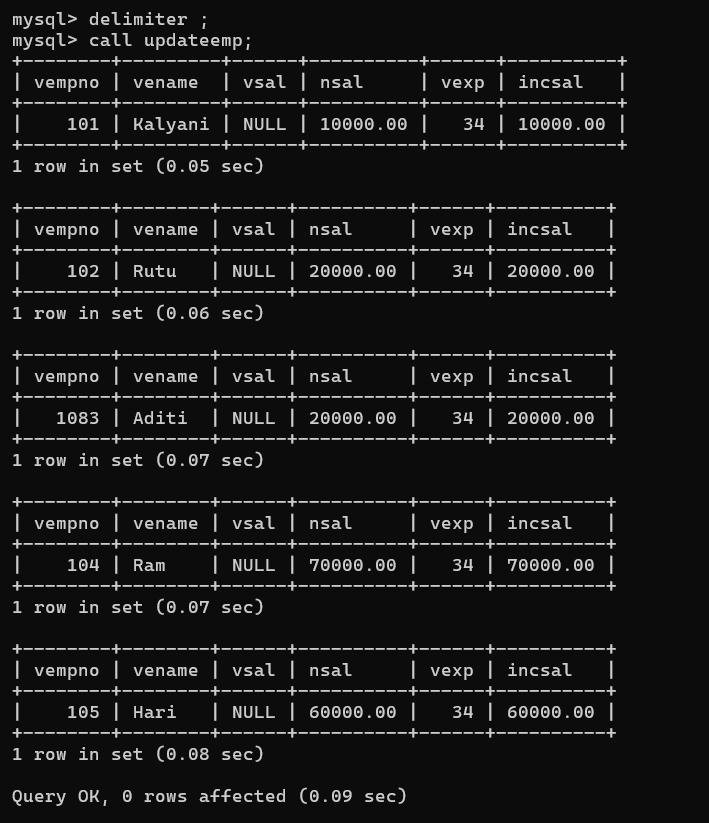


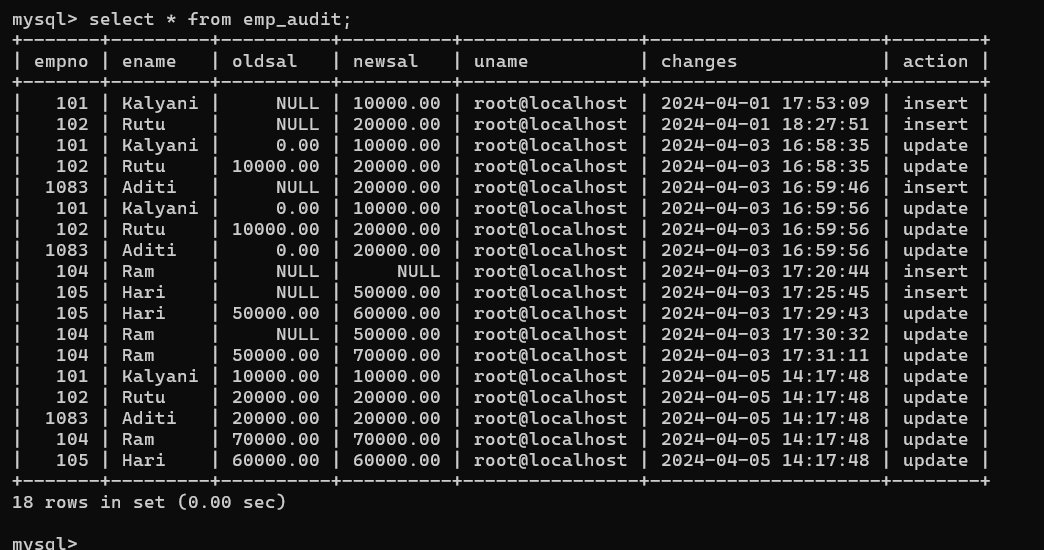


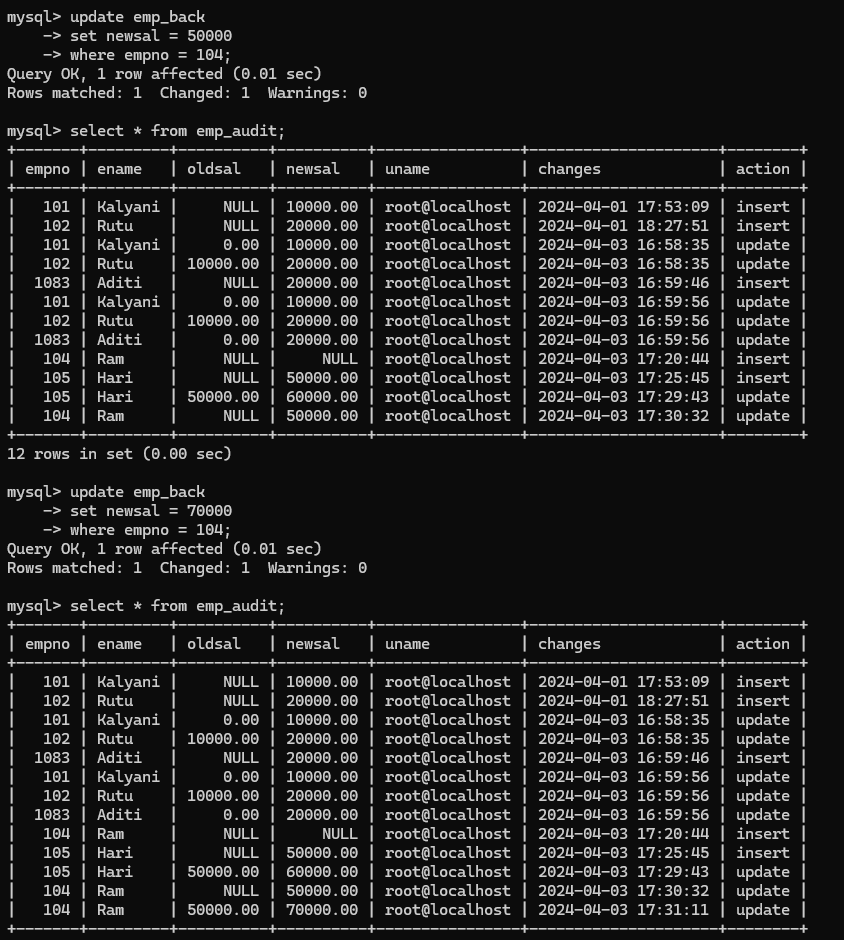












2. Write a trigger which add entry in audit table when user tries to insert or delete

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 trigger insertemp after insert on emp\_var

for each row

insert into empaudit values(

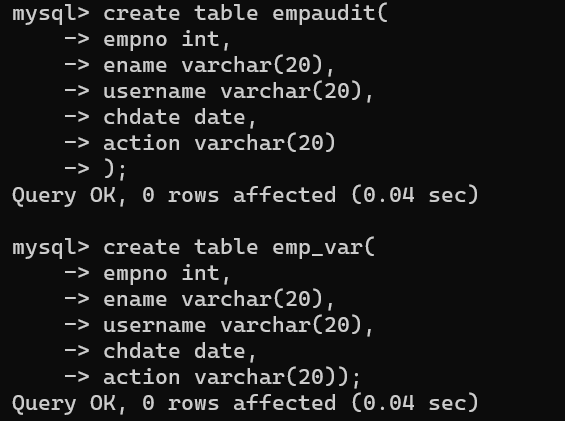
new.empno,new.ename,user(),now(),'INSERT');

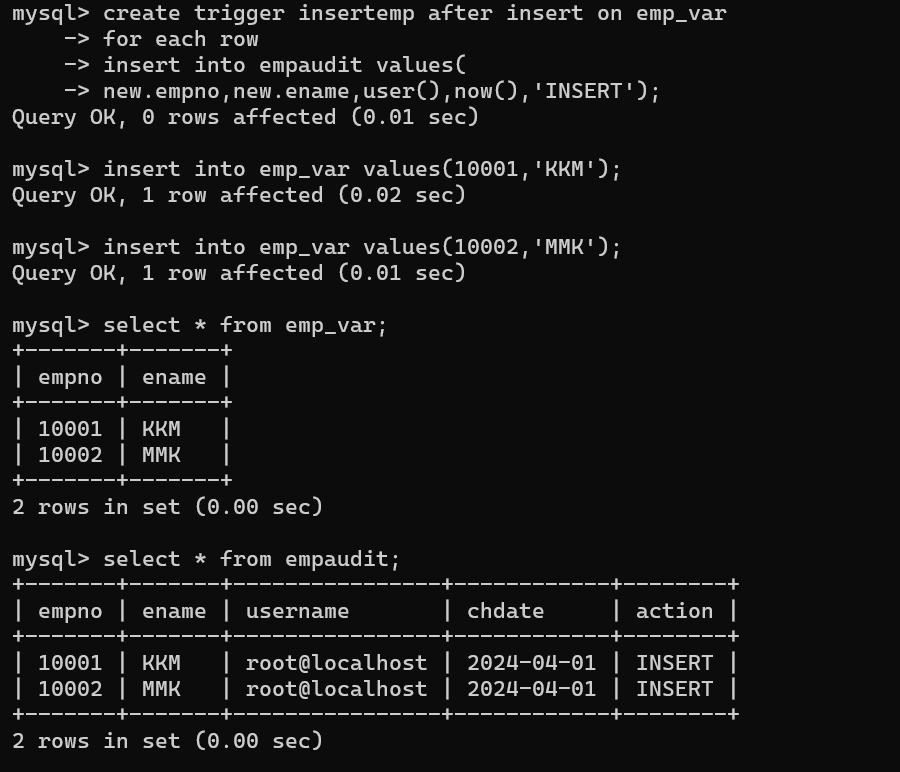
create trigger deleteemp after delete on emp\_var

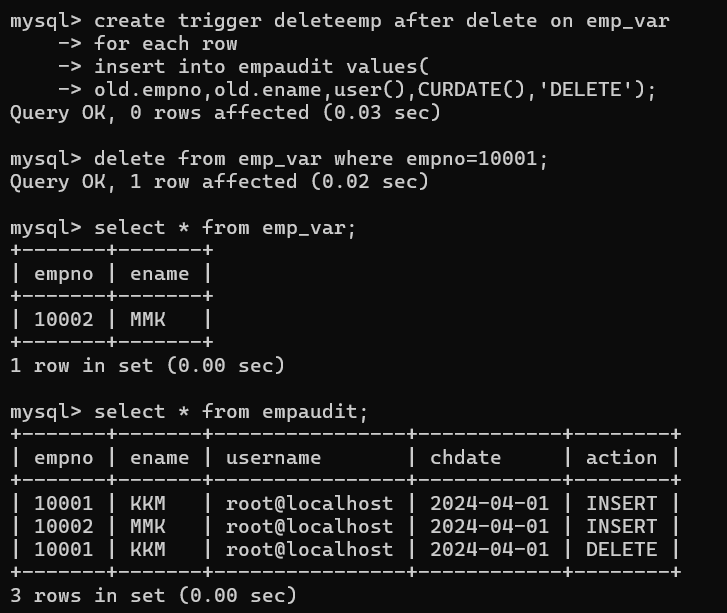
for each row

insert into empaudit values(

old.empno,old.ename,user(),CURDATE(),'DELETE');







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 Vehicle1(

vid int,

vname varchar(20),

price decimal(9,2),

chdate date,

username varchar(20),

action varchar(20));

create table vehicle\_history(

vid int,

vname varchar(20),

oldprice decimal(9,2),

newprice decimal(9,2),

chdate date,

username varchar(20),

action varchar(20));

create trigger insertveh after insert on vehicle1

for each row

insert into vehicle\_history values(

new.vid,new.vname,null,new.price,user(),now(),'INSERT');

create trigger updateveh after update on vehicle1

for each row

insert into vehicle\_history values(

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

create trigger deleteveh after delete on vehicle1

for each row

insert into vehicle\_history values(

old.vid,old.vname,old.price,CURDATE(),user(),'DELETE');

