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//

;

**Solution:**

delimiter //

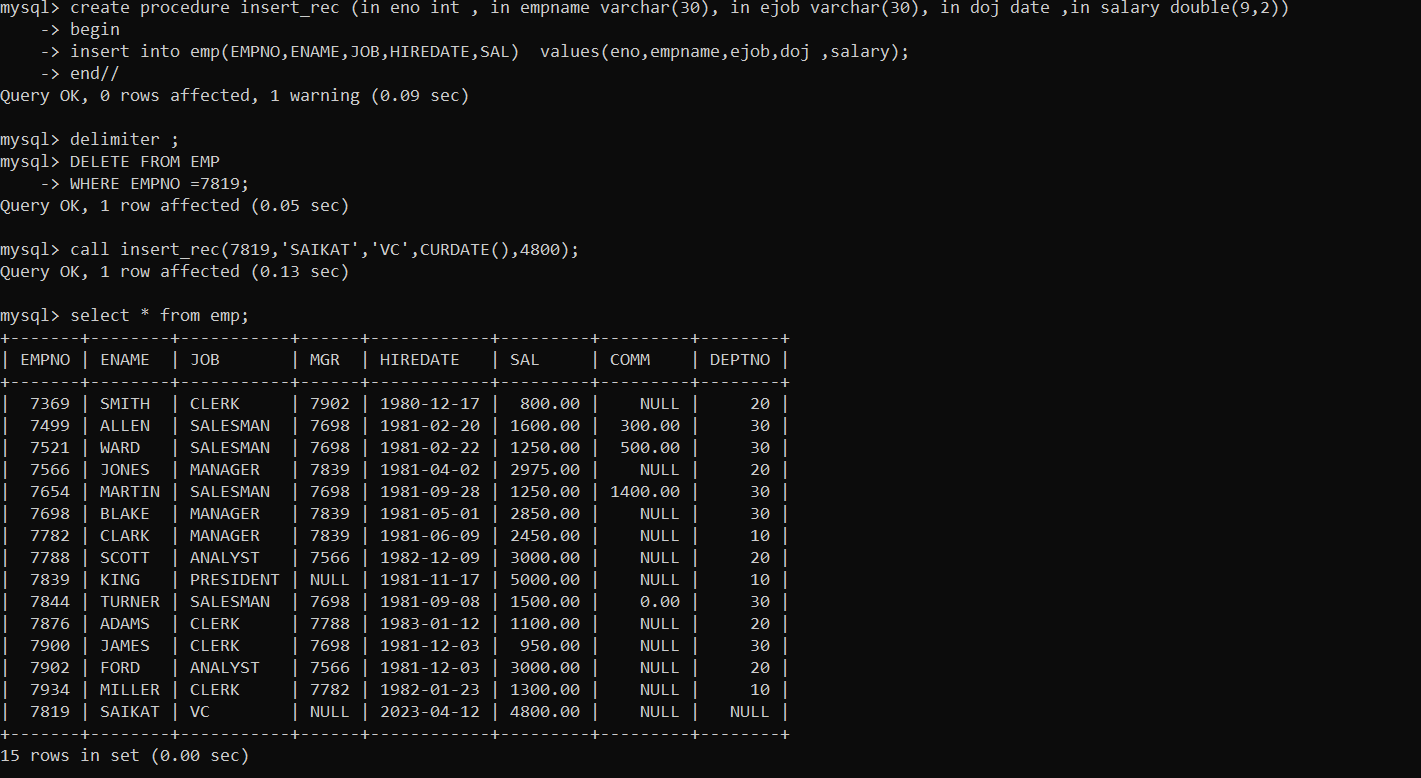
create procedure insert\_rec (in eno int , in empname varchar(30), in ejob varchar(30), in doj date ,in salary double(9,2))

begin

insert into emp(EMPNO,ENAME,JOB,HIREDATE,SAL) values(eno,empname,ejob,doj ,salary);

end//

delimiter ;



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

:

Delimiter //

create procedure delete\_emp(in eno int)

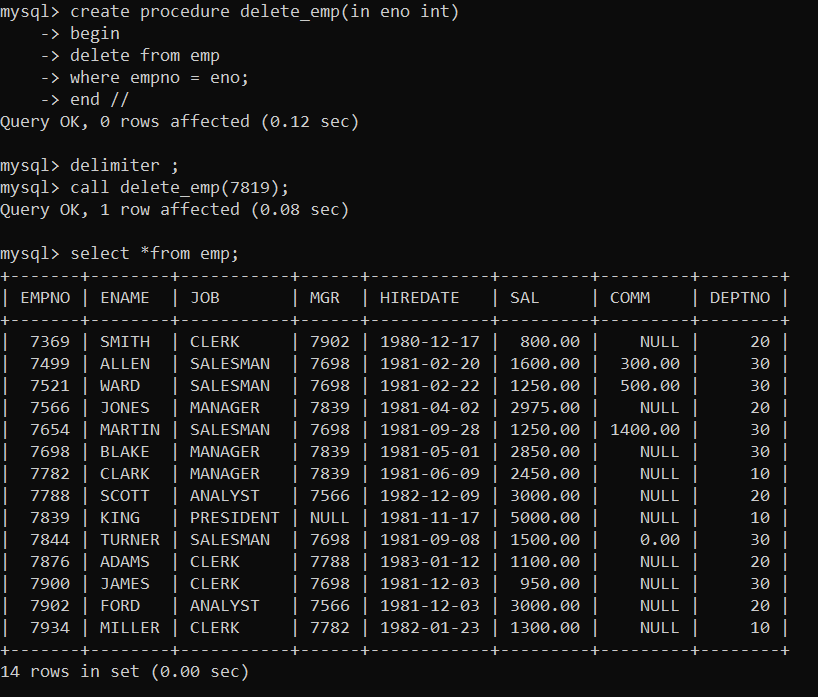
begin

delete from emp

where empno = eno;

end //

delimiter ;



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

> given salary. pass salary as a parameter to procedure

**Solution:**

Delimiter //

Create procedure display (in salary double (9,2))

Begin

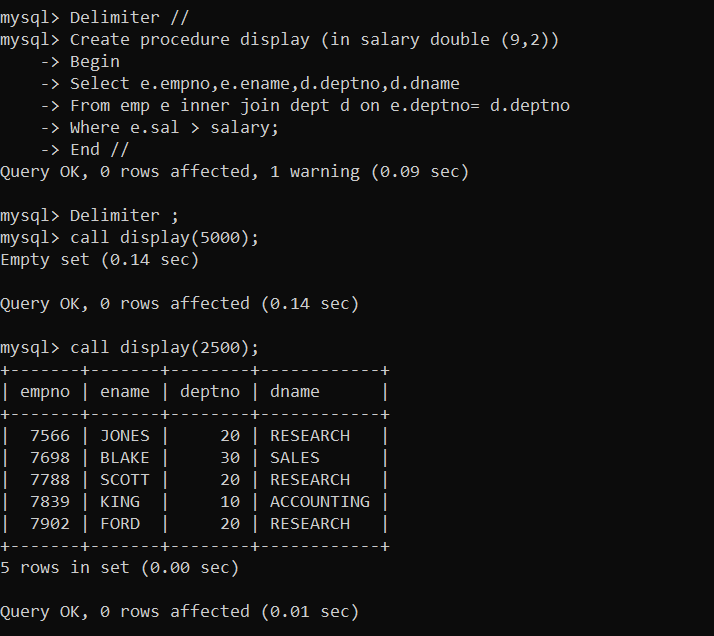
Select e.empno,e.ename,d.deptno,d.dname

From emp e inner join dept d on e.deptno= d.deptno

Where e.sal > salary;

End //

Delimiter ;



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

**Solution:**

delimiter //

create procedure get\_deptdetails(in dpno int,out max\_sal double(9,2), out min\_sal double(9,2), out avg\_sal double(9,2),out no\_of\_employees int )

begin

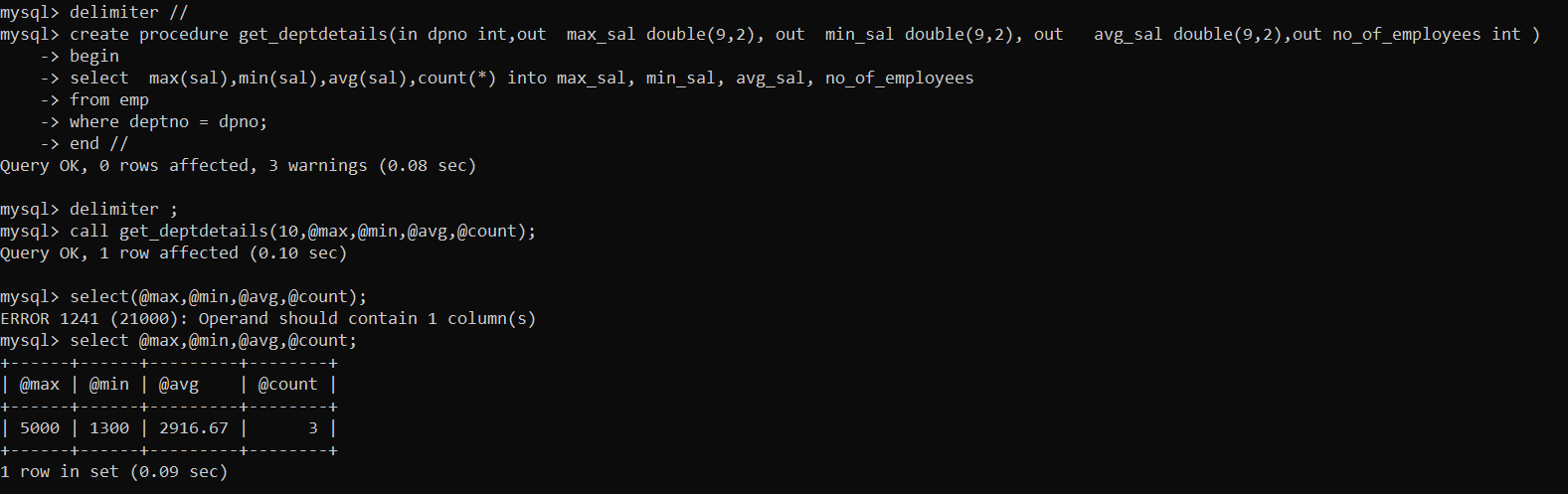
select max(sal),min(sal),avg(sal),count(\*) into max\_sal, min\_sal, avg\_sal, no\_of\_employees

from emp

where deptno = dpno;

end //

delimiter ;



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

product,category and salesman table)\

**Solution:**

delimiter //

create procedure disp\_allprod()

begin

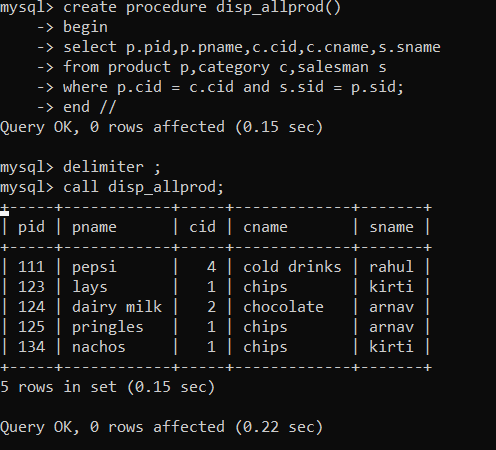
select p.pid,p.pname,c.cid,c.cname,s.sname

from product p,category c,salesman s

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

end //

delimiter ;



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

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

**Solution:**

delimiter //

create procedure show\_veh(in cuname varchar (30))

begin

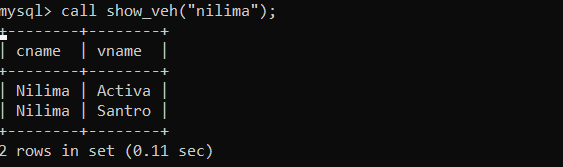
select c.cname,v.vname

from customer c ,vehicle v, cust\_vehicle cv

where c.cid =cv.cid and v.vid=cv.vid and c.cname = cuname;

end //

delimeter ;



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.

**Solution:**

Delimiter //

Create procedure emp\_info()

Begin

Declare finished int default 0;

Declare vempno , vdeptno int ;

Declare vename,vjob, vStatus varchar(30);

Declare vSal double(9,2);

declare vavg double(9,2);

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

Declare continue handler for not found set finished =1;

Open emp\_cur;

L1:loop

fetch emp\_cur into vempno,vename,vdeptno,vjob,vsal ;

if finished =1 then

leave L1;

end if;

select round(avg(sal),2) into vavg

from emp

where deptno = vdeptno;

if vsal > vavg then

set vstatus = “greater”;

elseif vsal< vavg then

set vstatus = “lesser”;

else

set vstatus =”equal”;

end if;

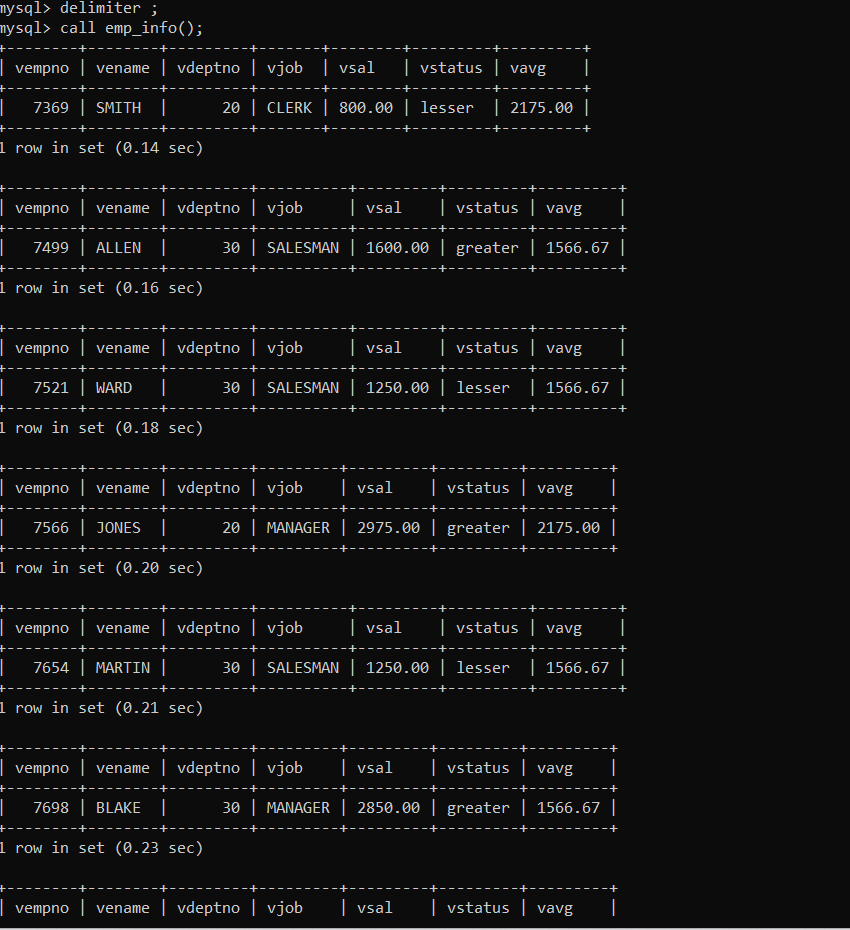
select vempno,vename,vdeptno,vjob,vsal ,vstatus,vavg;

end loop;

close emp\_cur;

End //

Delimiter ;



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

Experience< =35 then no Update

Experience> 35 and <=38 then 20% of salary

Experience> 38 then 25% of salary

**Solution :**

Delimiter //

Create procedure change\_sal()

Begin

Declare finished int default 0;

Declare experience,vempno int;

Declare mult float;

Declare vhiredate date;

Declare vsal double(9,2);

Declare emp\_cur cursor for Select empno,hiredate

From emp;

Declare continue handler for NOT FOUND set finished =1;

Open emp\_cur;

L11:loop

Fetch emp\_cur into vempno,vhiredate;

If finished =1 then

Leave L11;

End if;

Set experience = my\_exp(vhiredate);

If experience <= 40 then

Set mult = 1;

Elseif experience <= 41 then

Set mult = 1.2;

Else

Set mult = 1.25;

End if;

Update emp

Set sal = mult \* sal

Where empno =vempno;

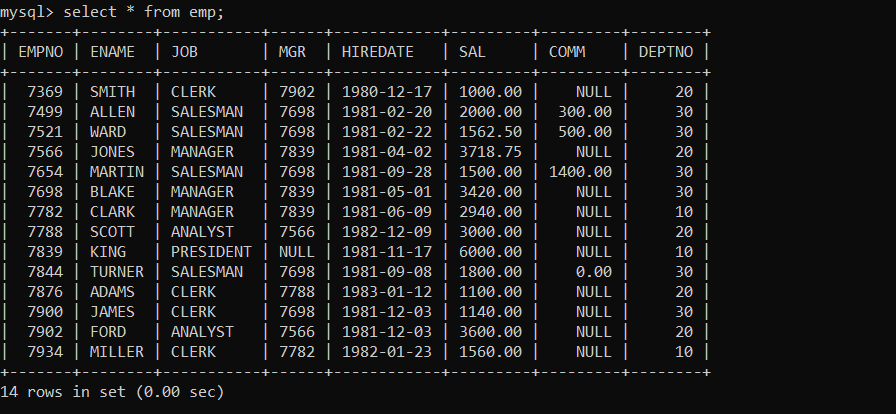
Select vempno,vhiredate,experience,mult;

End loop;

Close emp\_cur;

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));

**Solution:**

delimiter //

create function my\_exp(ehiredate date) returns int

begin

declare experience int;

set experience= floor((datediff (curdate(),ehiredate))/365);

return experience;

end //

DELIMITER ;

DELIMITER //

create procedure show\_allowance()

begin

declare finished int default 0;

declare vempno,vexperience int;

declare vallowance double(9,2);

declare vename varchar(20);

declare vhiredate date;

declare emp\_cur cursor for select empno,ename,hiredate from emp;

declare continue handler for NOT FOUND set finished =1;

open emp\_cur;

L12:loop

fetch emp\_cur into vempno,vename,vhiredate;

if finished =1 then

leave L12;

end if;

SET vexperience = my\_experience(vhiredate);

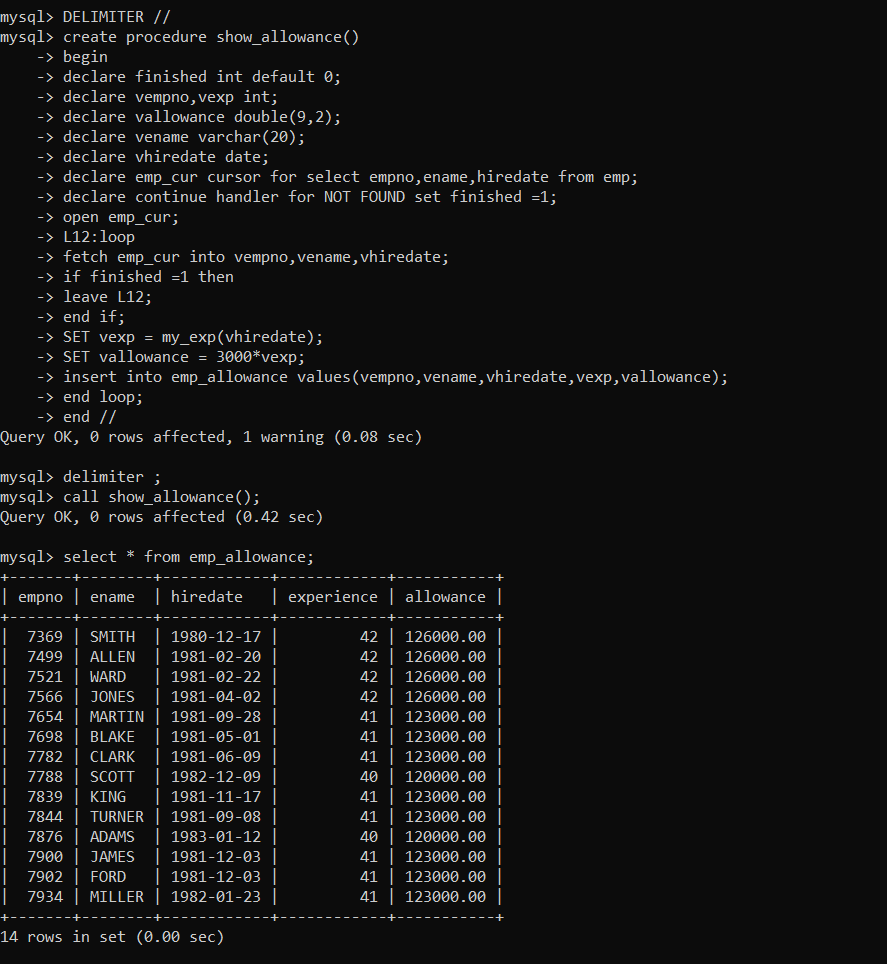
SET vallowance = 3000\*vexperience;

insert into emp\_allowance values(vempno,vename,vhiredate,vexperience,vallowance);

end loop;

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.

< 30 Year Nil

>=30 Year< 35 Year 10% of Salary

>=35 Year< 40 Yea……………………………….r 20% of Salary

>40 Year 30% of Salary

**Solution:**

Delimiter //

Create function cost(salary double(9,2),ehiredate date) returns double(9,2)

Begin

Declare DA double(9,2) default 0.15 \*salary;

Declare HRA double(9,2) default 0.2 \*salary;

Declare TA double(9,2) default 0.08 \*salary;

Declare spl\_allw,cost double(9,2);

Declare experience int ;

Declare mult float;

Set experience = my\_experience(ehiredate);

If experience between 30 and 35 then

Set mult =0.1;

Elseif experience<30 then

Set mult =0;

elseIf experience <= 40 then

Set mult= 0.2;

Else

Set mult = 0.3;

End if;

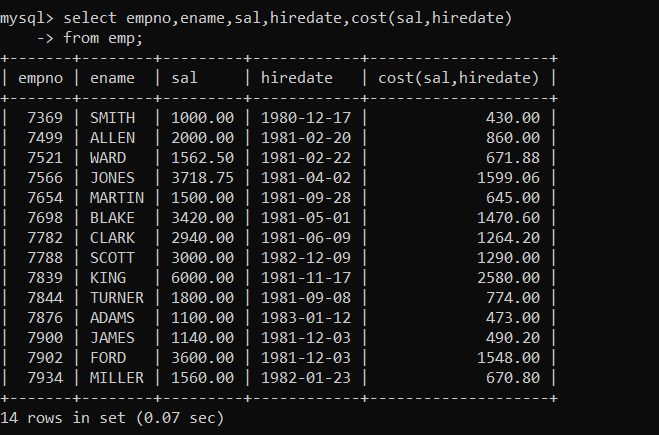
Set spl\_allw= mult \*salary;

Set cost = spl\_allw + TA +DA + HRA;

Return cost;

End //

Delimiter ;



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

use function written in question 10)

**Solution:**

delimiter //

create procedure display\_cost()

begin

declare finished int default 0;

declare vempno int;

declare vsal,vcost double(9,2);

declare vename varchar(30);

declare vhiredate date;

declare emp\_cur cursor for select empno,ename, sal,hiredate from emp;

declare continue handler for NOT FOUND set finished =1;

open emp\_cur;

L11:loop

fetch emp\_cur into vempno,vename,vsal,vhiredate;

if finished =1 then

leave L11;

end if;

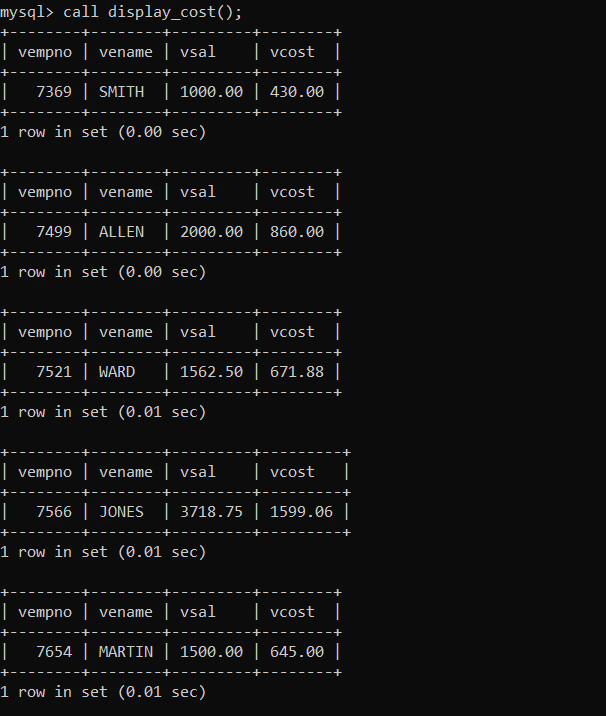
set vcost = cost (vsal,vhiredate);

select vempno,vename,vsal,vcost;

end loop;

end //

delimiter ;



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(

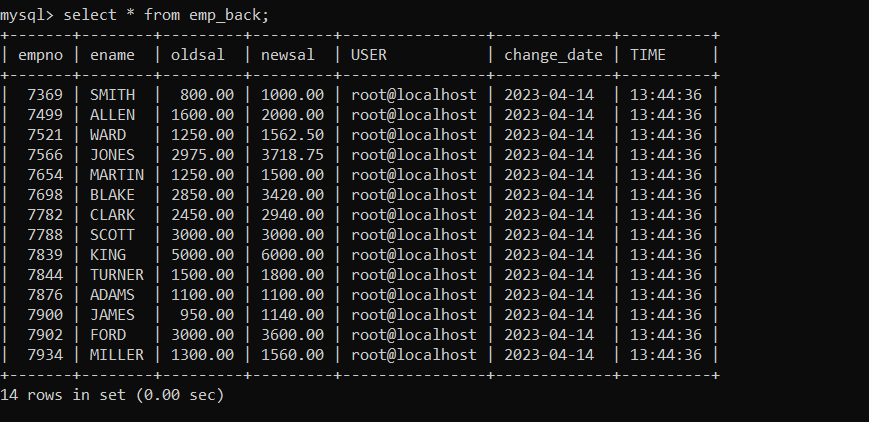
empno int,

ename varchar(20),

oldsal decimal(9,2),

newsal decimal(9,2)

);



**Solution:**

create table emp\_back(

empno int,

ename varchar(20),

oldsal decimal(9,2),

newsal decimal(9,2)

);

Delimiter //

Create trigger update\_log before update

on emp For each row

Begin

Insert into emp\_back values(old.empno,old.ename,old.sal,new.sal,user(),curdate(),curtime());

End //

Delimiter ;

(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)

);

**Solution:**

create table empaudit(

empno int,

ename varchar(20),

username varchar(20),

chdate date,

action varchar(20)

);

Delimiter //

Create trigger emp\_delete before delete on emp

For each row

Begin

Insert into empaudit values (old.empno,old.ename,user(),curdate(),”delete”);

End //

Create trigger emp\_insert after insert on emp

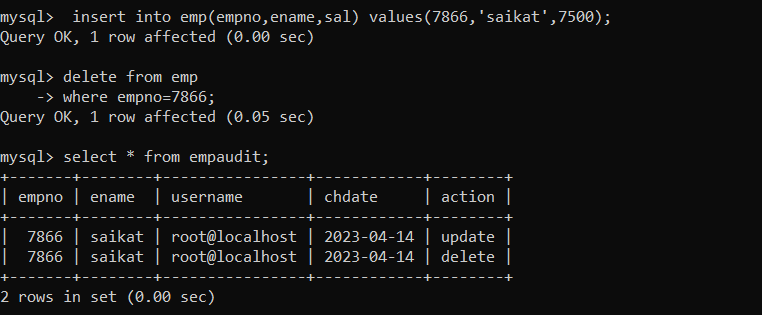
For each row

Begin

Insert into empaudit values (new.empno,new.ename,user(),curdate(),”update”);

End //

Delimiter ;



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)

);

**Solution:**

create table vehicle\_history(

vno int,

vname varchar(20),

oldprice double(9,2),

newprice double(9,2),

chdate date,

chtime time,

username varchar(20)

);

Delimiter //

Create trigger veh\_update before update on vehicle

For each row

Begin

Insert into vehicle\_history values (old.vid,old.vname,old.price,new.price,curdate(),curtime(),user());

End //

Delimiter ;

