**PL-SQL ASSIGNMENT**

**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

**delimiter //**

**create procedure insert\_rec(vempno int, vename varchar(20), vsal decimal(7,2), vjob varchar(20),vhiredate date)**

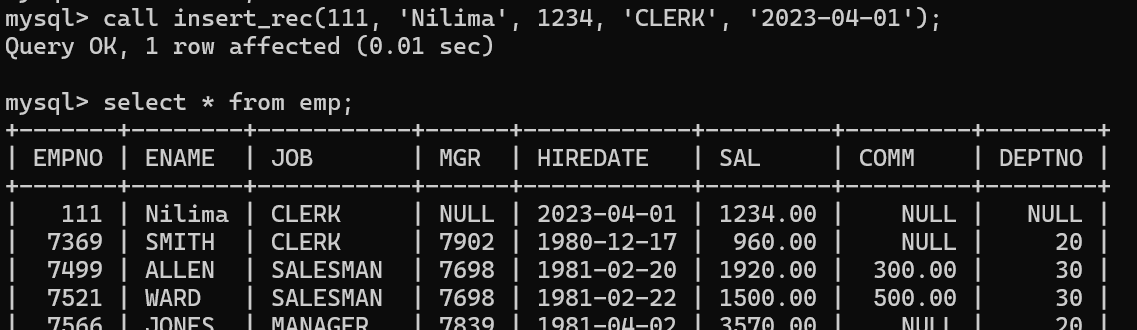
**begin**

**insert into emp values(vempno, vename, vjob, null, vhiredate, vsal, null, null);**

**end //**

**delimiter ;**

**call insert\_rec(111, 'Nilima', 1234, 'CLERK', '2023-04-01');**



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\_value(in eno int)**

**begin**

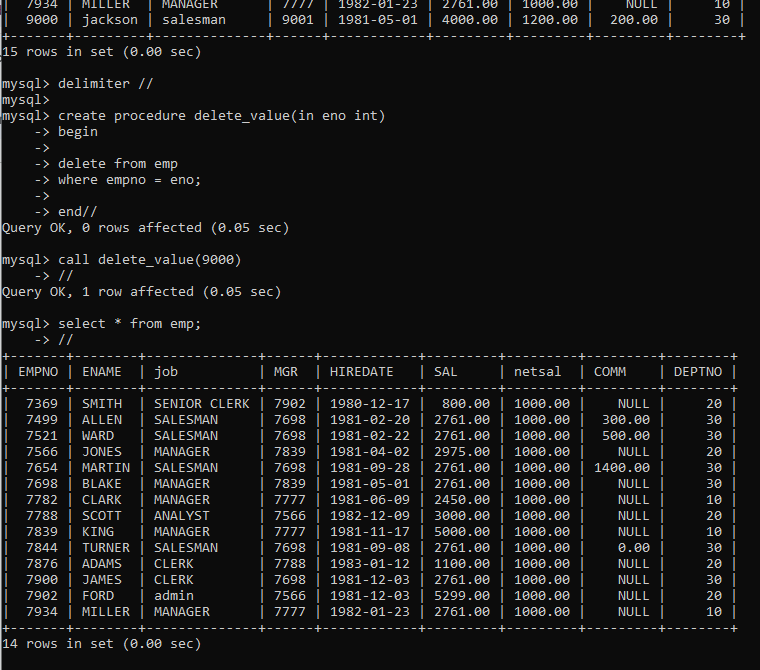
**delete from emp**

**where empno = eno;**

**end//**

**delimiter ;**

**call delete\_value(9000);**



3. write a procedure to display empno,ename,deptno,dname for all employees with sal > given salary. pass salary as a parameter to procedure

**delimiter //**

**create procedure display\_emp(in esal int)**

**begin**

**select empno, ename, dept.deptno, dept.dname**

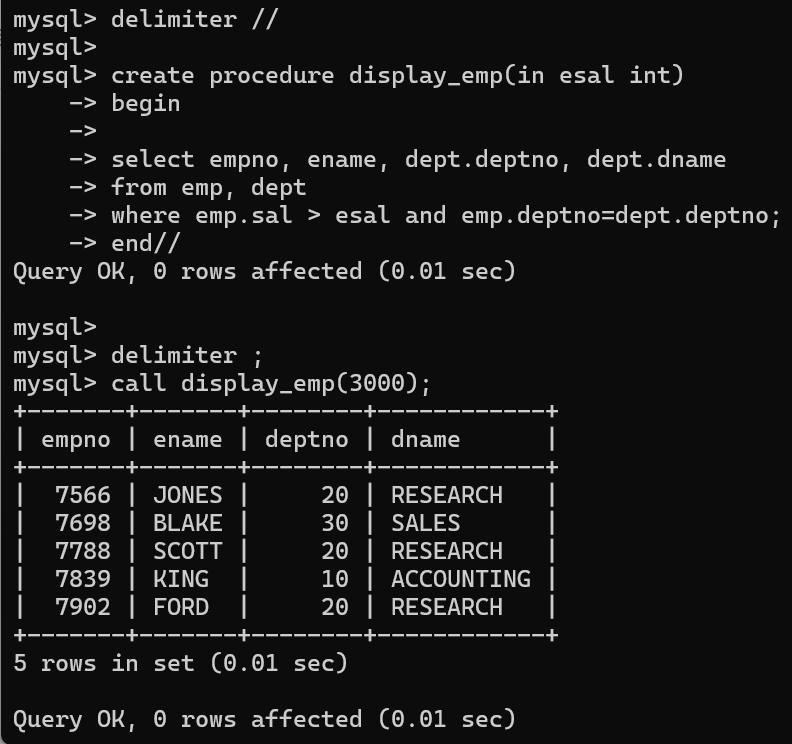
**from emp, dept**

**where emp.sal > esal and emp.deptno=dept.deptno;**

**end//**

**delimiter ;**

**call display\_emp(3000);**

****

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

**delimiter //**

**create procedure emp\_details(in dno int, out mi float(7,2), out ma float(7,2), out av float(7,2), out ct float(7,2))**

**begin**

**select min(sal), max(sal), avg(sal), count(empno) into mi, ma, av, ct**

**from emp**

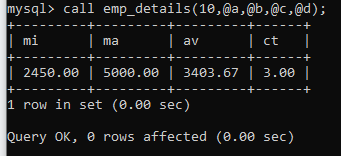
**where deptno = dno;**

**select mi, ma, av, ct;**

**end//**

**delimiter ;**

**call emp\_details(10,@a,@b,@c,@d);**

****

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

**salesman(sid, sname, city)**

**product(prodid,pname,qty,price,catid,sid)**

**category(cid, cname, description)**

**delimiter //**

**create procedure displayproduct()**

**begin**

**select p.prodid, p.pname, p.qty, p.price, s.sname, s.sid, s.city, c.cid, c.cname, c.description**

**from product as p**

**Left join salesman s on p.sid=s.sid**

**Left join category as c on p.catid=c.cid;**

**end//**

**delimiter ;**

----LEFT JOIN FOR 3 TABLES

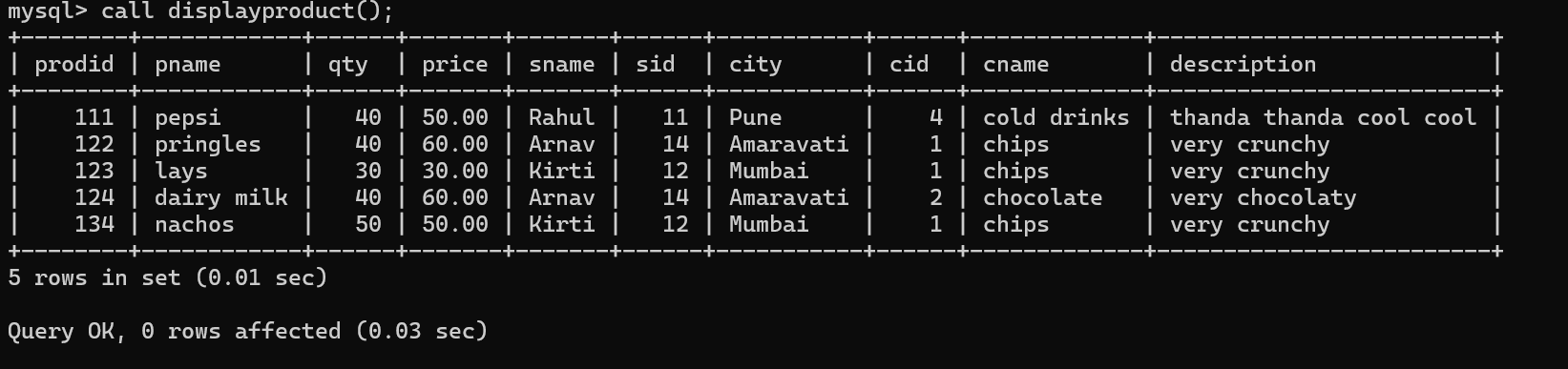
**SELECT t1.column1, t2.column2, t3.column3**

**FROM table1 AS t1**

**LEFT JOIN table2 AS t2 ON t1.id = t2.id**

**LEFT JOIN table3 AS t3 ON t1.id = t3.id**

**WHERE t1.condition = 'value';**



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(cust\_vehicle))

vehicle(vid, vname, price, description)

customer(custid, cname, address)

cust\_vehicle(custid, vid, sid, buy\_price)

**delimiter //**

**create procedure displaycustvehicle(in custname varchar(20))**

**begin**

**select v.vid, v.vname, v.price, v.description, c.custid, c.cname, c.address**

**from vehicle v , customer c , cust\_vehicle as cv**

**where c.cname=custname and**

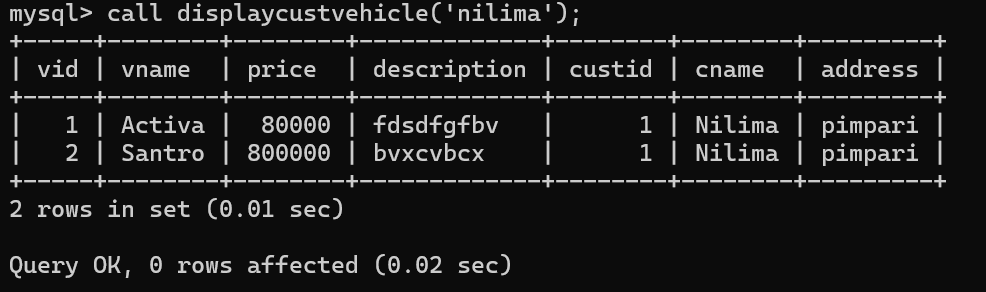
**cv.custid=c.custid and**

**cv.vid=v.vid;**

**end //**

**delimiter ;**

**call displaycustvehicle('nilima');**

****

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 vename,vjob,vstatus varchar(20);**

**declare vsal,vavgsal float(9,2);**

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

**declare continue handler for NOT FOUND set vfinished=1;**

**#load the data from table to cursor**

**open empcur;**

**label1:loop**

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

**if vfinished =1 then**

**leave label1;**

**end if;**

**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;**

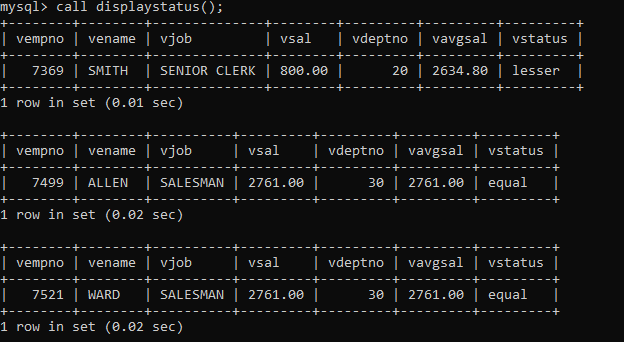
**end loop;**

**close empcur;**

**end//**

**delimiter ;**

**Call displaystatus();**

****

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

Exp< =41 then no Update

Exp> 41 and <=43 then 20% of salary

Exp> 43 then 25% of salary

**delimiter //**

**create procedure updatesal()**

**begin**

**declare vfinished, exp int default 0;**

**declare vempno, vincsal int;**

**declare vename , updatesal varchar(20);**

**declare vsal float(9,2);**

**declare vhiredate date;**

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

**declare continue handler for NOT FOUND**

**set vfinished=1;**

**OPEN empcur;**

**label1:loop**

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

**if vfinished=1 then**

**leave label1;**

**end if;**

**select floor(datediff(curdate(),vhiredate)/365) into exp;**

**if exp > 41 then**

**set vincsal=vsal\*1.2;**

**update emp**

**set sal = vincsal**

**where empno=vempno;**

**else**

**set vincsal=vsal\*1.25;**

**update emp**

**set sal = vincsal**

**where empno=vempno;**

**end if;**

**select vempno,vename,vhiredate,vsal,vincsal;**

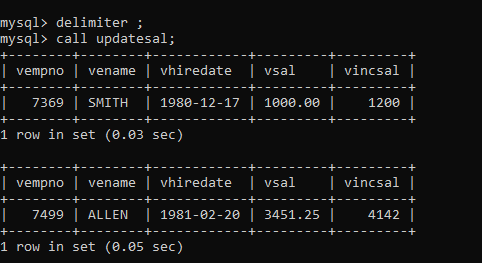
**end loop;**

**close empcur;**

**end//**

**delimiter ;**

**Call updatesal;**

****

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));

**--FUNCTION**

**delimiter //**

**create function calexp(vhiredate date) returns int**

**begin**

**declare vexp int;**

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

**return vexp;**

**end //**

**delimiter ;**

**--TABLE**

**create table emp\_allowance(**

**empno int,**

**ename varchar(20),**

**hiredate date,**

**experience int,**

**allowance decimal(10,2)**

**);**

**--PROCEDURE**

**delimiter //**

**create procedure empallowance()**

**begin**

**declare vfinished int default 0;**

**declare vempno,vexp int;**

**declare vename varchar(20);**

**declare vhiredate date;**

**declare vallowance decimal(10,2);**

**declare allowancecur cursor for select empno, ename, hiredate from emp;**

**declare continue handler for NOT FOUND set vfinished=1;**

**open allowancecur;**

**label1:loop**

**fetch allowancecur into vempno, vename, vhiredate;**

**if vfinished =1 then**

**leave label1;**

**end if;**

**set vexp=calexp(vhiredate);**

**set vallowance=vexp\*3000;**

**#select vempno, vename, vhiredate, vexp, vallowance;**

**insert into emp\_allowance(empno, ename, hiredate, experience, allowance) values(vempno, vename, vhiredate, vexp, vallowance);**

**end loop;**

**close allowancecur;**

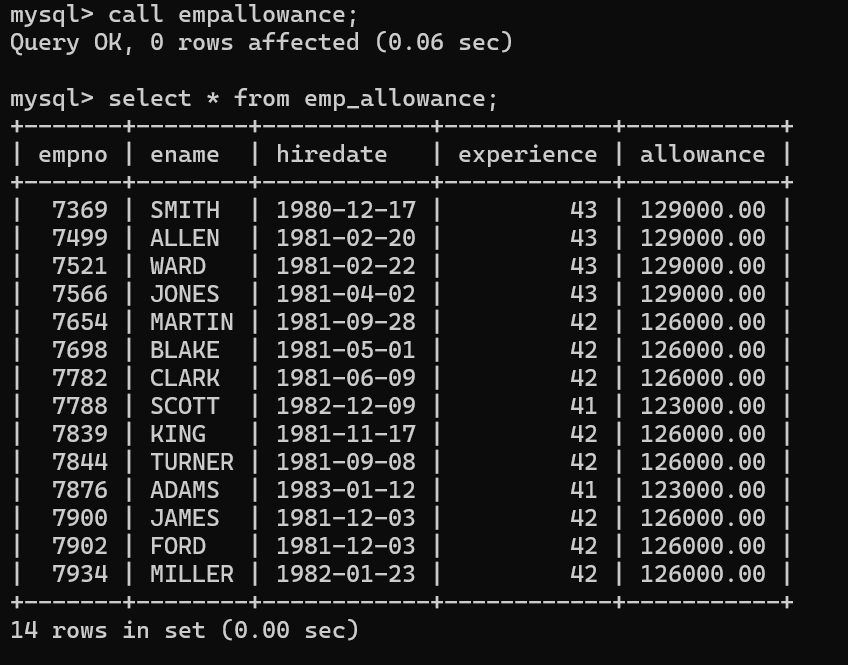
**end //**

**delimiter ;**

**—---------CALL PROCEDURE**

**Call empallowance;**

**select \* from emp\_allowance;**

****

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

**--FUNCTION**

**delimiter //**

**create function calCTC(vsal decimal(7,2), vhiredate date) returns decimal(7,2)**

**begin**

**declare vda, vhra, vta decimal(7,2);**

**declare vsa decimal(7,2) default 0;**

**declare vgrosssal decimal(7,2);**

**declare vexp int;**

**declare allowance varchar(20);**

**set vda=0.15\*vsal;**

**set vhra=0.2\*vsal;**

**set vta=0.8\*vsal;**

**set vgrosssal = vsal+vda+vhra+vsa+vta;**

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

**if vexp < 41 then**

**set allowance='No Special Allowance' ;**

**elseif vexp < 42 then**

**#set allowance='10% Allowance' ;**

**set vgrosssal=vgrosssal\*1.1;**

**elseif vexp < 44 then**

**#set allowance='20% Allowance';**

**set vgrosssal=vgrosssal\*1.2;**

**else**

**#set allowance='30% Allowance';**

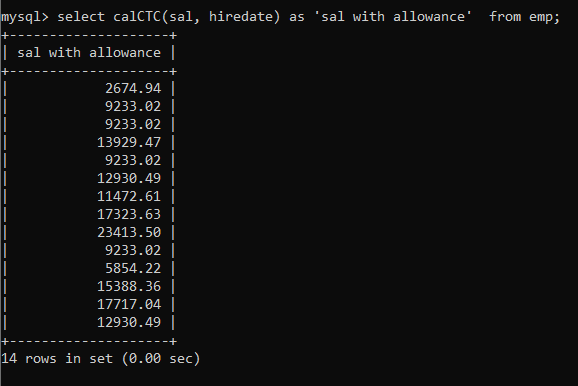
**set vgrosssal=vgrosssal\*1.3;**

**end if;**

**return vgrosssal;**

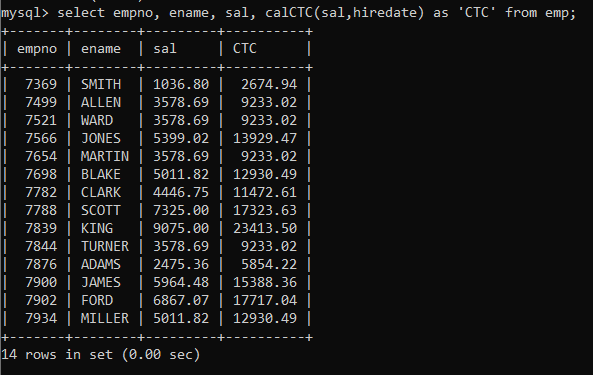
**end //**

**delimiter ;**

****

11. Write query to display empno,ename,sal,cost to company for all employees(note: use function written in question 10)

**select empno, ename, sal, calCTC(sal,hiredate) as 'CTC' from emp;**



**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)

**------TABLE**

**create table emp\_back(**

**empno int,**

**ename varchar(20),**

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

**newsal decimal(9,2),**

**oldjob varchar(20),**

**newjob varchar(20),**

**uname varchar(20),**

**changes datetime,**

**action varchar(20)**

**)**

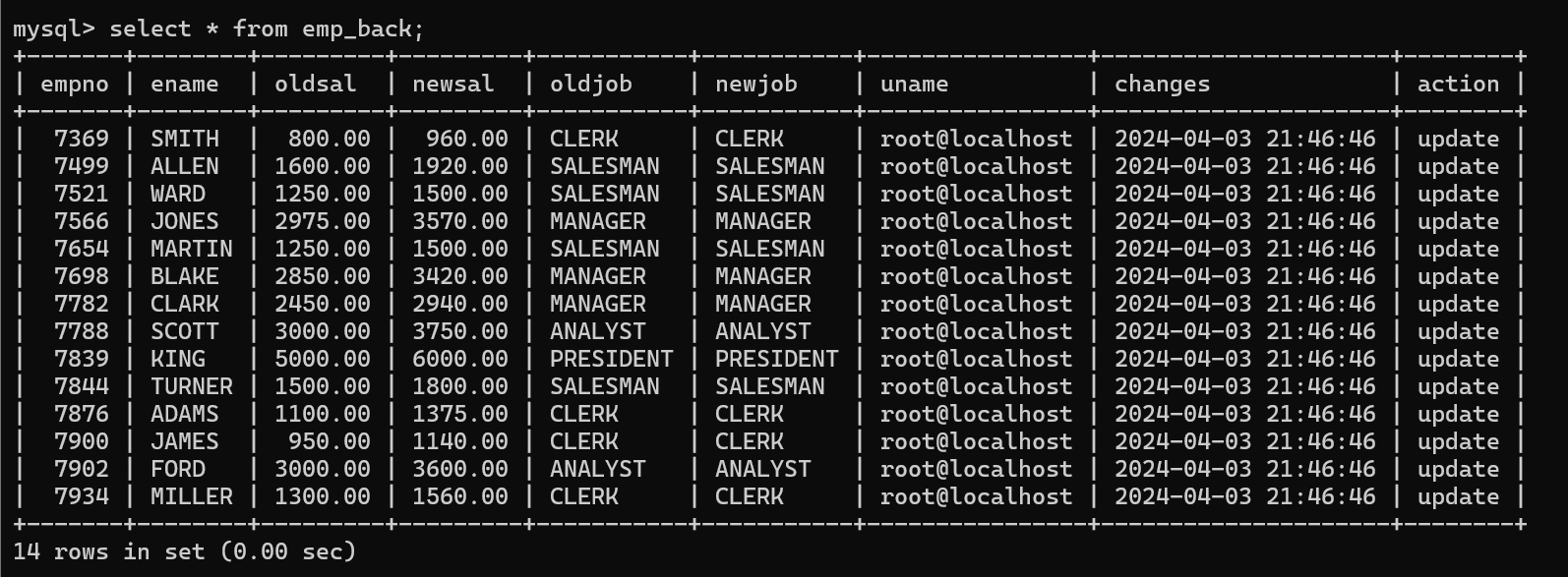
**---UPDATE TRIGGER  
create trigger updateemp before update on emp for each row**

**insert into emp\_back values(old.empno, old.ename, old.sal, new.sal, old.job, new.job, user(), now(), 'update');**

--Then after running the procedure in the question number 8 the values in the emp\_back gets updated

**Displaying the values in the emp\_back table**

**Select \* from emp\_back;**



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 emp\_audit(

empno int,

oldename varchar(20),

newename varchar(20),

salllll decimal(9,2),

newsal decimal(9,2),

uname varchar(20),

changes datetime,

action varchar(20)

)

--DELETE TRIGGER

create trigger deleteemp after delete on emp for each row

insert into emp\_back values(

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

--INSERT TRIGGER

create trigger insertemp after insert on emp for each row

insert into emp\_audit values(old.empno, old.ename, new.ename, old.sal, new.sal, user(), now(),'insert');

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)

);

—---------VEHICLE TABLE

**create table vehicle(**

**vno int primary key,**

**vname varchar(20),**

**price decimal(9,2)**

**);**

**insert into vehicle values(1,'Car',1234);**

**insert into vehicle values(2,'Bike',1234);**

**insert into vehicle values(3,'Car 2',1234);**

**insert into vehicle values(4,'abcd',1234);**

**insert into vehicle values(5,'cdcc',1234);**

—---------VEHICLE HISTORY TABLE

**--TABLE**

**create table vehicle\_history(**

**vno int,**

**vname varchar(20),**

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

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

**uname varchar(20),**

**changes datetime,**

**action varchar(20)**

**);**

—---------UPDATE TRIGGER

**create trigger updatevrhicle before update on vehicle for each row**

**insert into vehicle\_history values(old.vno, old.vname, old.price, new.price, user(), now(), 'update');**

—----UPDATING value in Vehicle table

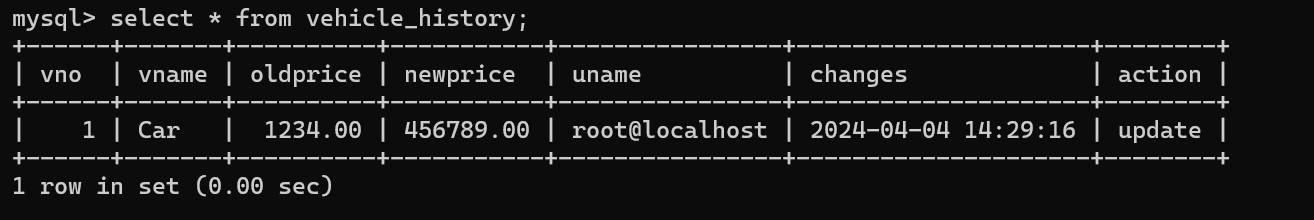
**update vehicle**

**set price=456789**

**where vno=1;**

—-----------------------------------

**Select \* from vehicle\_history;**

****