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 //

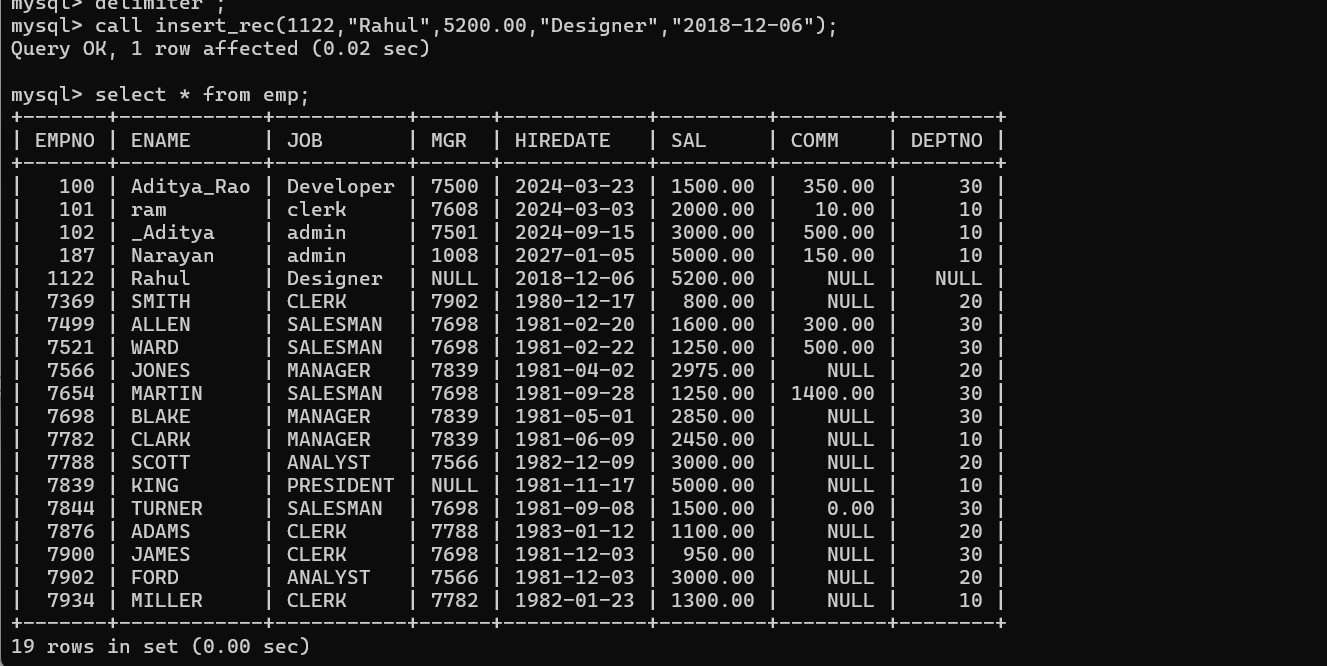
create procedure insert\_rec(pempno int ,pename varchar(25),psal float(9,2),pjob varchar(25),phiredate date) begin

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

values(pempno,pename,psal,pjob,phiredate); end //

delimiter ;

call insert\_rec(1122,"Rahul",5200.00,"Designer","2018-12-06");



1. 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

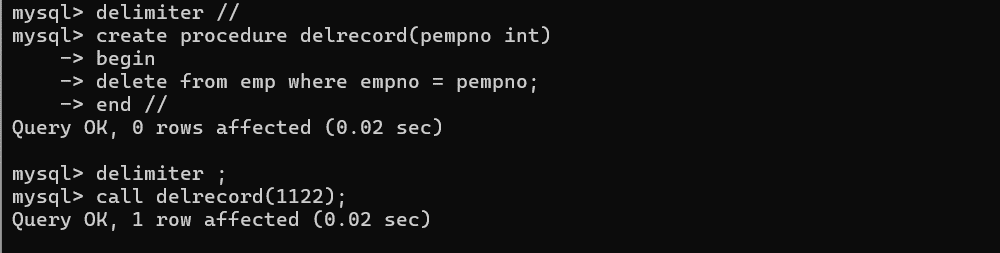
delimiter //

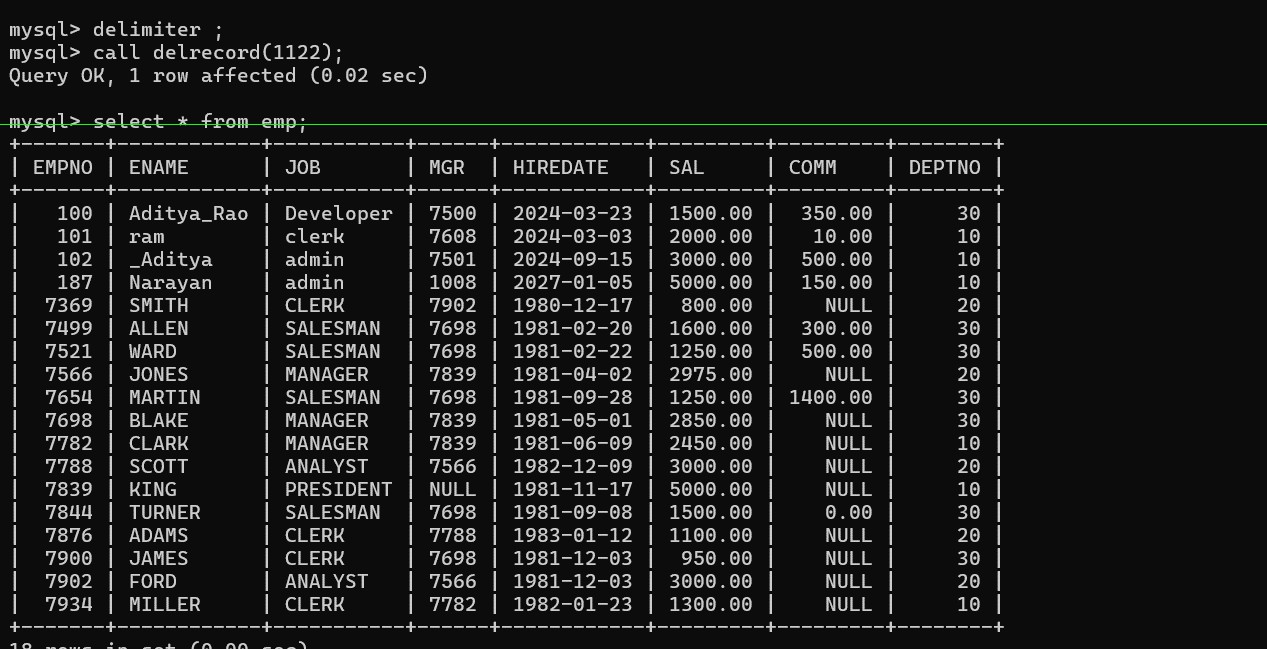
create procedure delrecord(pempno int) begin

delete from emp where empno = pempno; end //

delimiter ;

call delrecord(1122);





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

> given salary. pass salary as a parameter to procedure

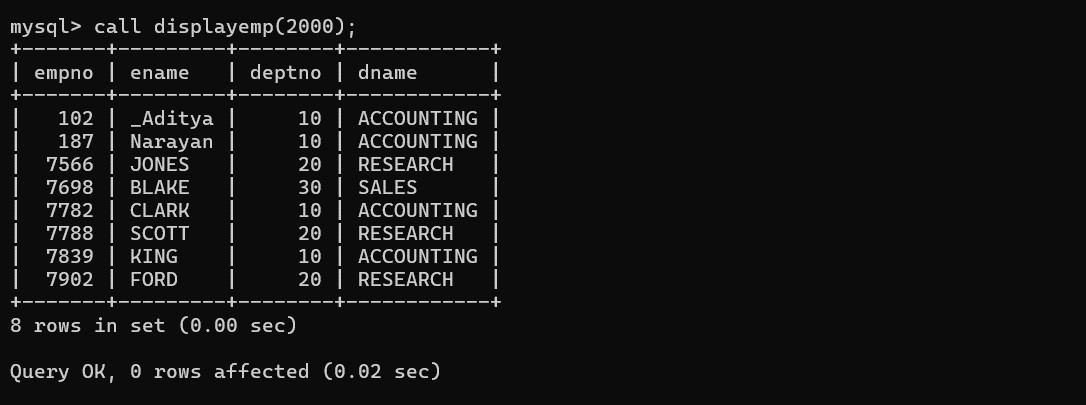
delimiter &&

create procedure displayemp(psal float(9,2)) begin

select empno,ename,d.deptno,d.dname from emp e,dept d where sal > psal and e.deptno = d.deptno;

end &&

delimiter ;



1. 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 Ans=>

delimiter %%

create procedure findminmax(in pdept int) begin

select max(sal),min(sal),avg(sal),count(\*) from emp where pdept = deptno;

end %% delimiter ;



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

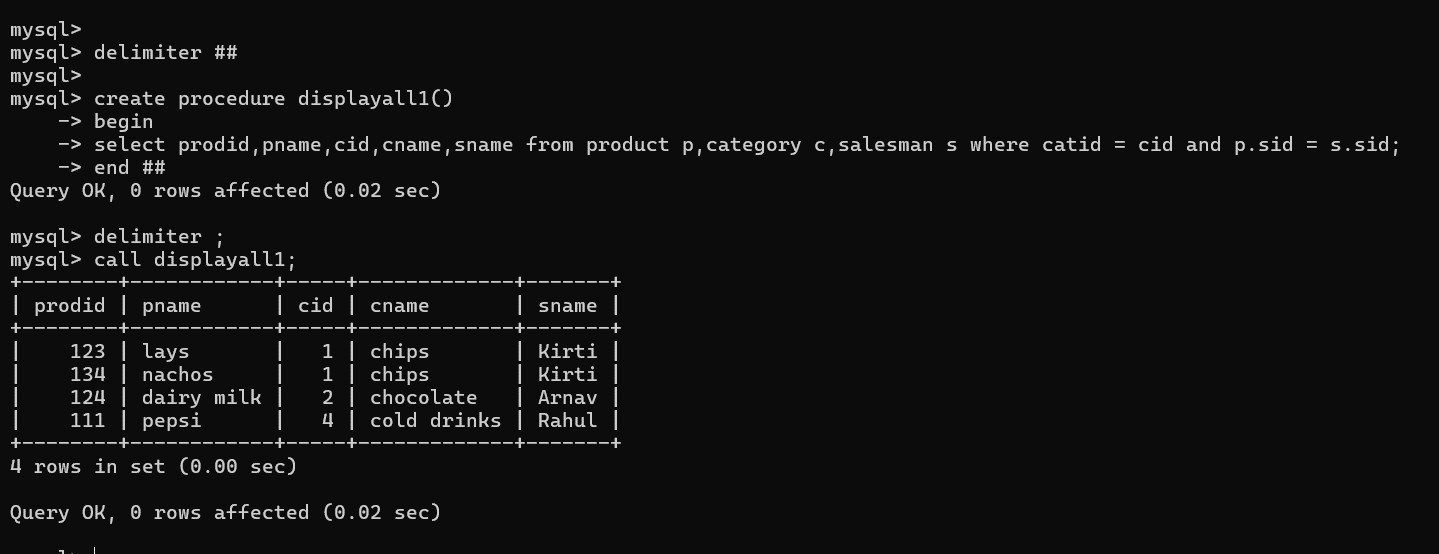
delimiter ##

create procedure displayall1() begin

select prodid,pname,cid,cname,sname from product p,category c,salesman s where catid = cid and p.sid = s.sid; end ##

delimiter ;

call displayall1();



1. 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 disinfo() begin

declare vempno, vdeptno, vfinished int default 0; declare vename, vstatus, vjob varchar(50) default ''; declare vsalary, vsal float(9,2);

declare empcur cursor for select empno, ename, sal,job, deptno from emp; declare continue handler for NOT FOUND set vfinished = 1;

open empcur; label1:

loop

fetch empcur into vempno, vename, vsal, vjob,vdeptno; if vfinished = 1 then

leave label1; end if;

select avg(sal) into vsalary from emp where deptno = vdeptno; if vsal > vsalary then

set vstatus = 'Greater'; elseif vsal < vsalary then

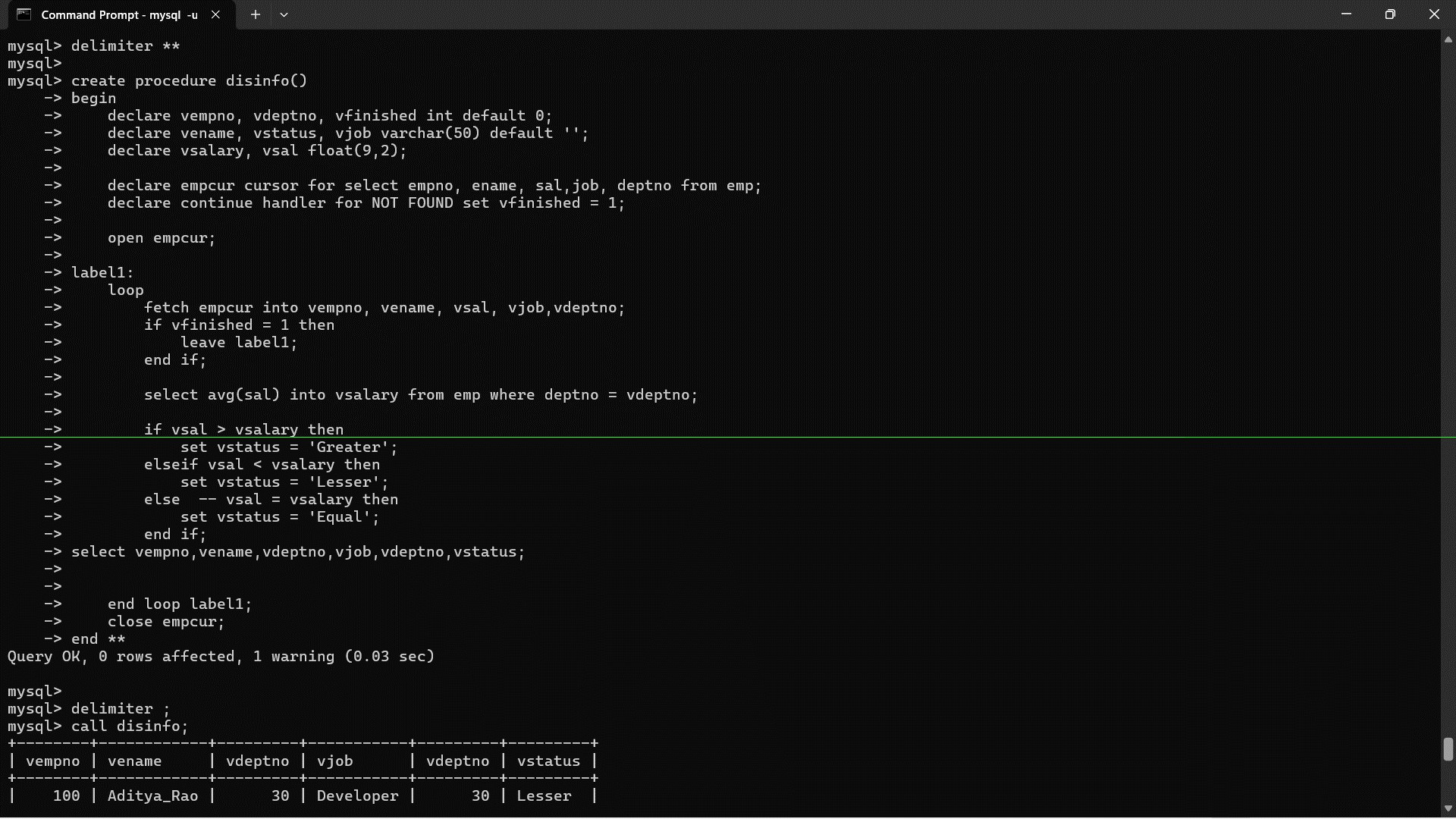
set vstatus = 'Lesser'; else -- vsal = vsalary then

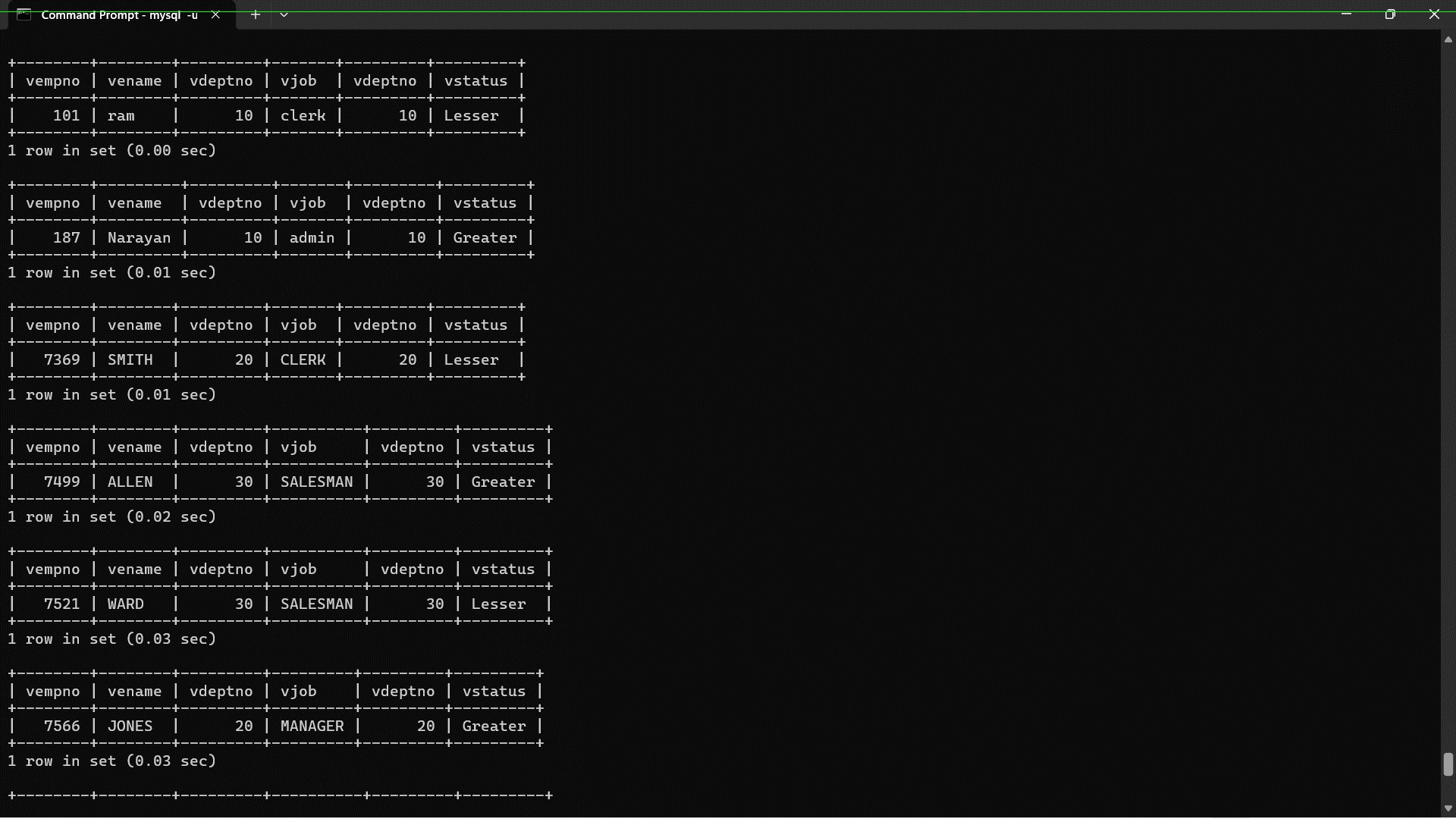
set vstatus = 'Equal'; end if;

select vempno,vename,vdeptno,vjob,vdeptno,vstatus; end loop label1;

close empcur; end \*\*

delimiter ;





1. 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. Ans =>

delimiter \*\*

create procedure empupdate1() begin

declare vexp, vfinished, vdeptno int default 0; declare vsal,oldsalary float(9,2);

declare vename, vjob varchar(50) default ''; declare vhiredate date;

declare vempno int;

declare empcur cursor for select empno, ename, job, deptno,sal, sal, hiredate from emp; declare continue handler for NOT FOUND set vfinished = 1;

open empcur;

label1:

loop

fetch empcur into vempno, vename, vjob, vdeptno,oldsalary, vsal, vhiredate; if vfinished = 1 then

leave label1; end if;

set vexp = calexp(vhiredate);

if vexp > 40 then

set vsal = vsal \* 1.25; elseif vexp > 38 then

set vsal = vsal \* 1.20; else

set vsal = vsal;

-- No change needed end if;

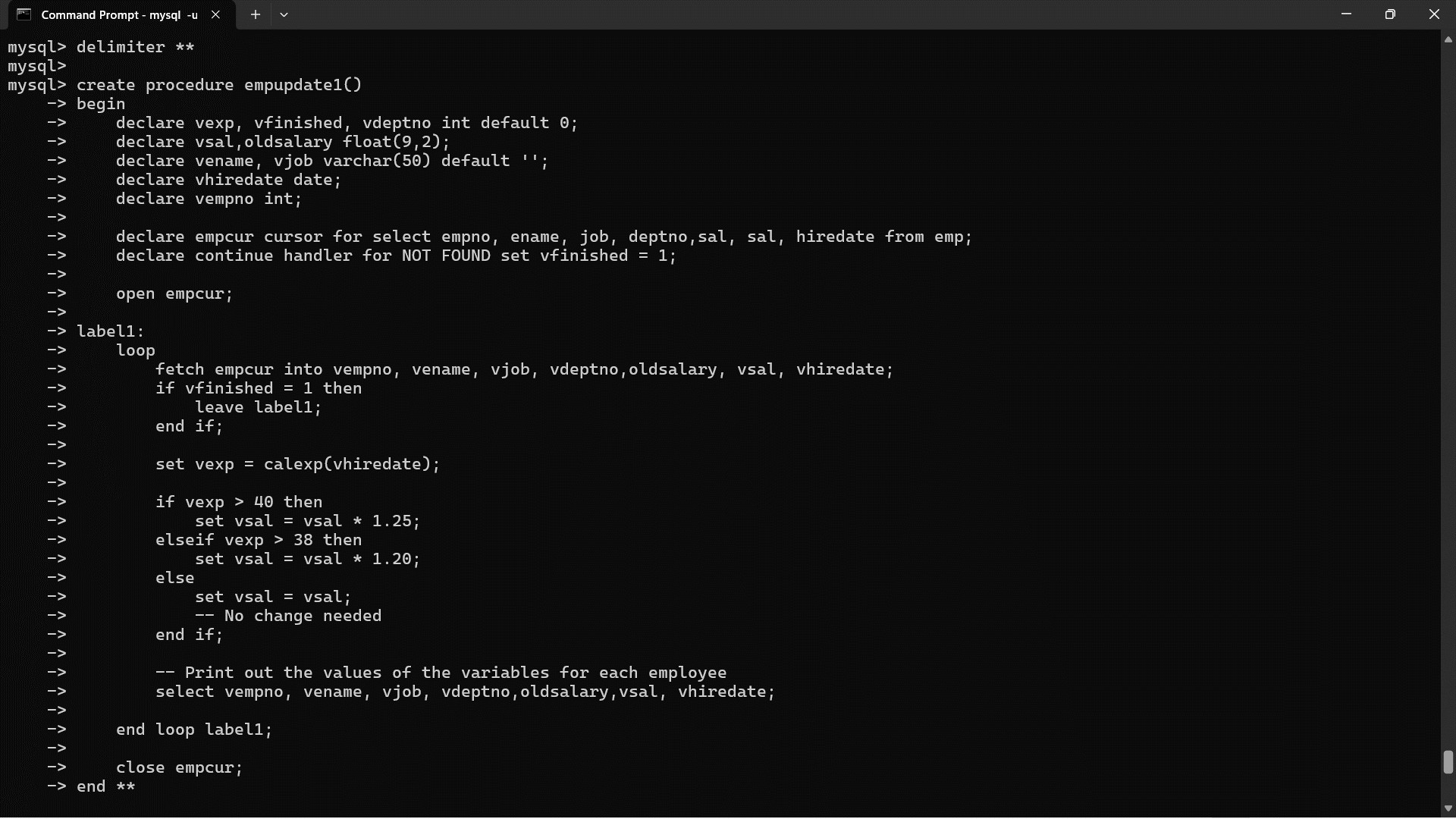
-- Print out the values of the variables for each employee

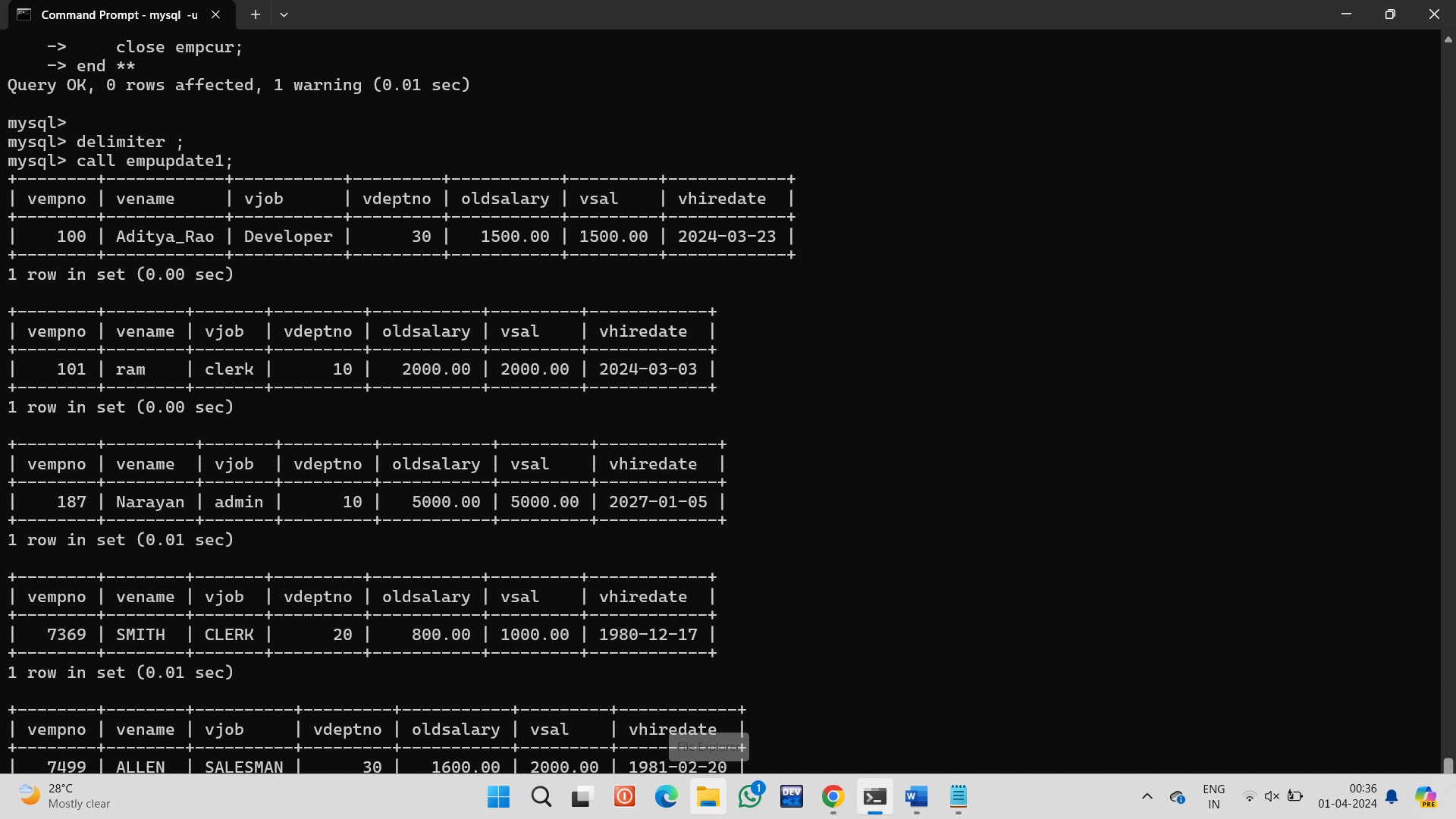
select vempno, vename, vjob, vdeptno,oldsalary,vsal, vhiredate;

end loop label1;

close empcur; end \*\*

delimiter ;





1. 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 empupdate2() begin

declare vexp, vfinished, vdeptno int default 0; declare vsal,additonal\_allowances float(9,2); declare vename, vjob varchar(50) default ''; declare vhiredate date;

declare vempno int;

declare empcur cursor for select empno, ename, job, deptno, sal, hiredate from emp; declare continue handler for NOT FOUND set vfinished = 1;

open empcur;

label1:

loop

fetch empcur into vempno, vename, vjob, vdeptno, vsal, vhiredate; if vfinished = 1 then

leave label1; end if;

set vexp = calexp(vhiredate);

set additonal\_allowances = vexp \* 3000;

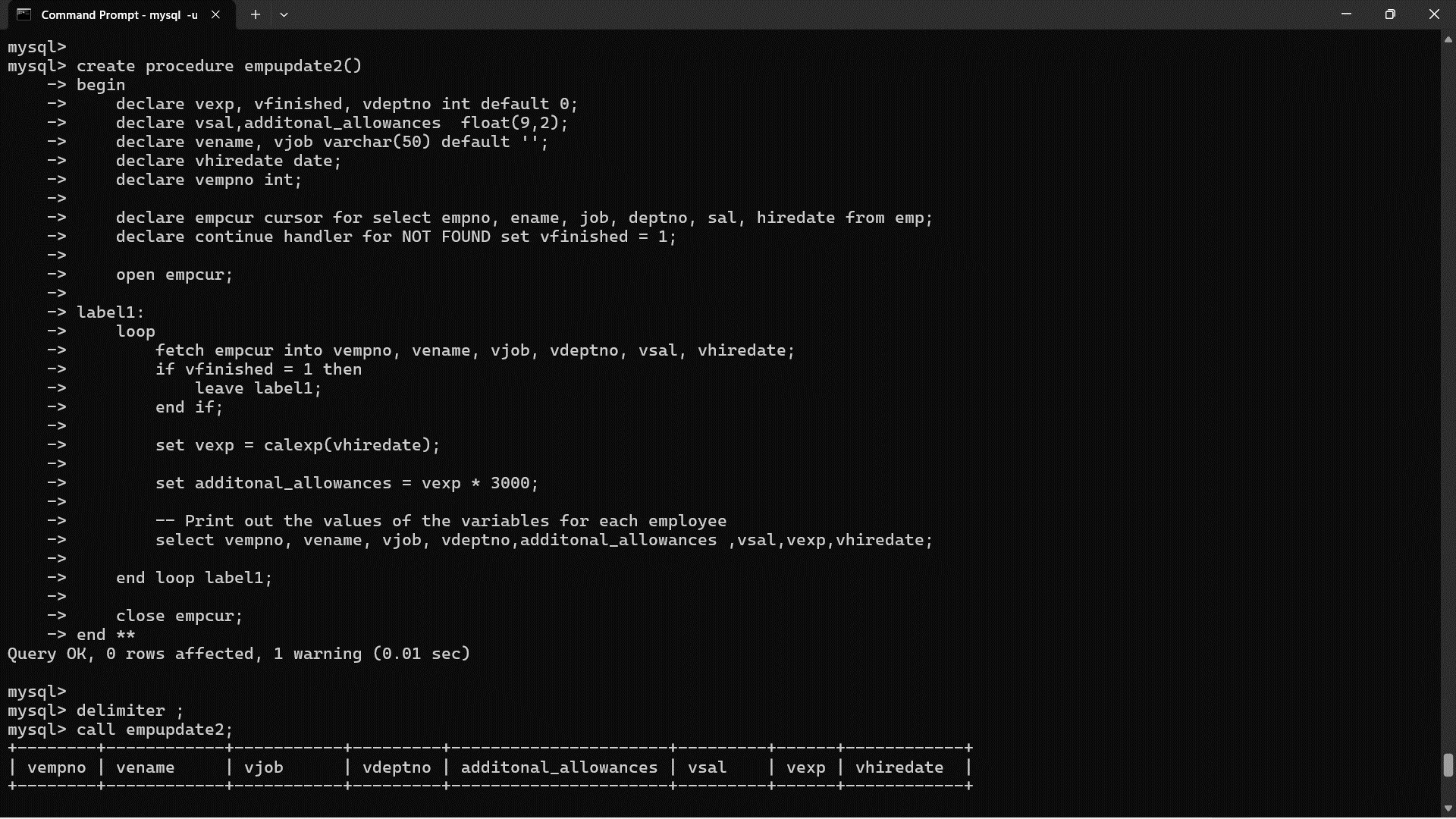
-- Print out the values of the variables for each employee

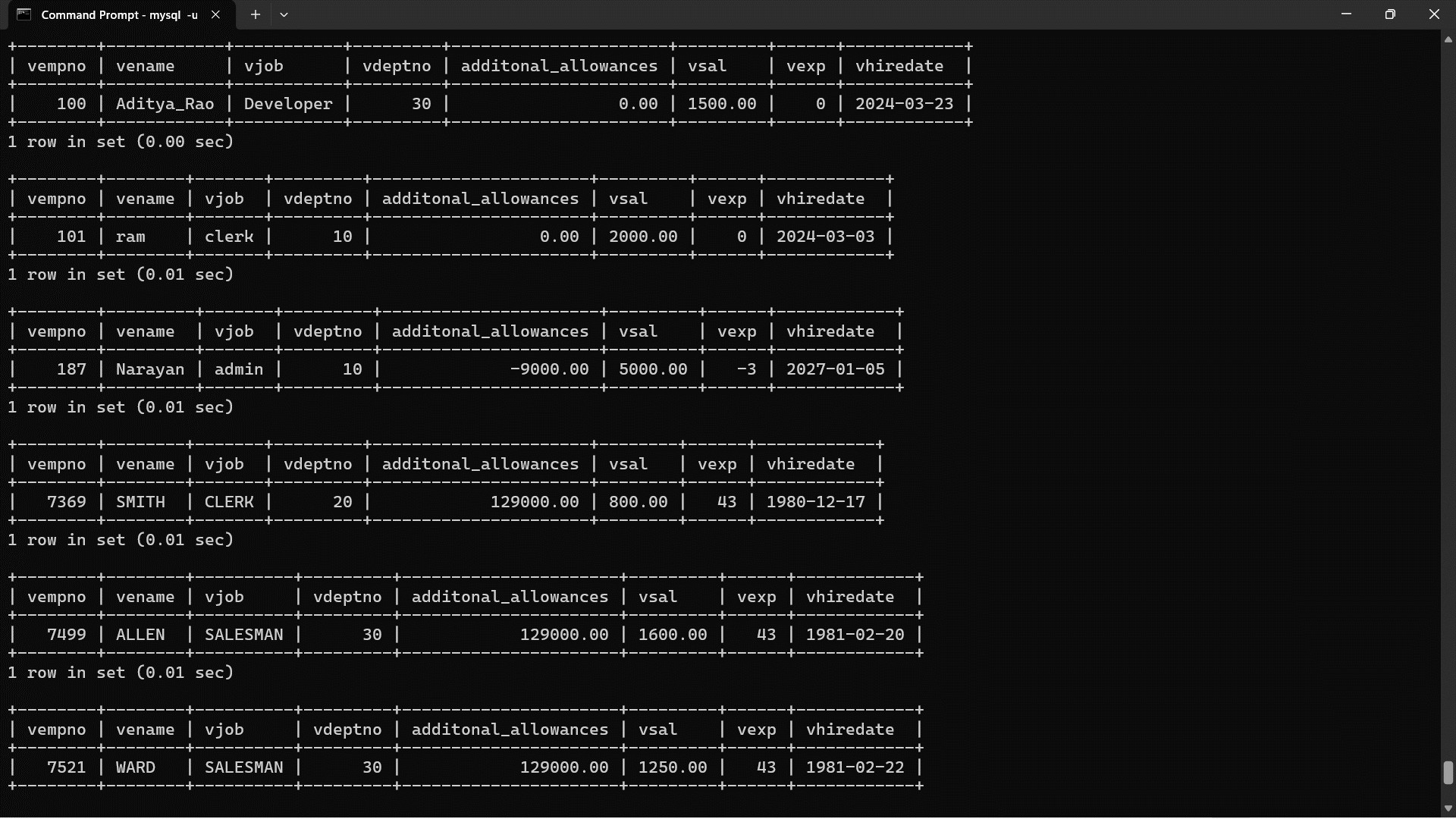
select vempno, vename, vjob, vdeptno,additonal\_allowances ,vsal,vexp,vhiredate;

end loop label1;

close empcur; end \*\*

delimiter ;





1. 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

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) Ans=>

create table emp\_back(empno int, ename varchar(20),

oldsal float(9,2), newsal float(9,2), user varchar(20), time date,

action varchar(20)

);

create trigger emptrigger before insert on emp for each row

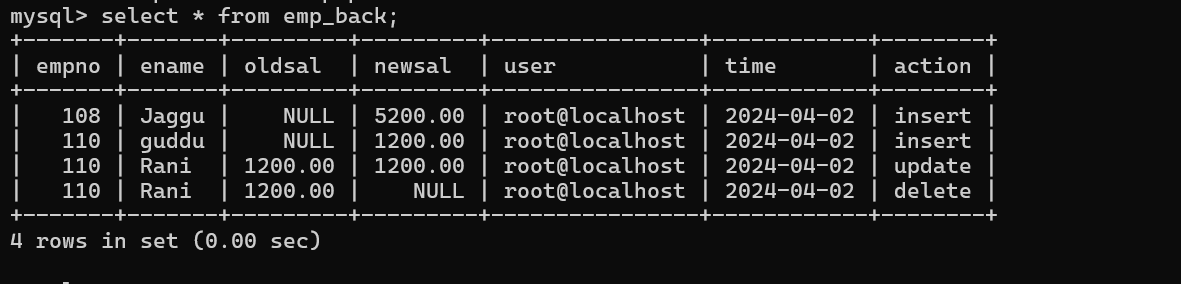
insert into emp\_back(empno,ename,oldsal,newsal,user,time,action) values(new.empno,new.ename,null,new.sal,user(),now(),'insert');

create trigger emptrigger1 before update on emp for each row insert into emp\_back(empno,ename,oldsal,newsal,user,time,action)

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

create trigger emptrigger2 before delete on emp for each row

insert into emp\_back(empno,ename,oldsal,newsal,user,time,action) values(old.empno,old.ename,old.sal,null,user(),now(),'delete');



1. 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)

);

Ans=>

create table empaudit(empno int, ename varchar(20),

user varchar(20), chdate date, action varchar(20)

);

create trigger insertemp before insert on emp for each row insert into empaudit(empno,ename,user,chdate,action) values(new.empno,new.ename,user(),now(),'insert');

create trigger delemp before delete on emp for each row insert into empaudit(empno,ename,user,chdate,action) values(old.empno,old.ename,user(),now(),'delete');



1. 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)

);

Ans=>

create table vehicle(vid int primary key,vname varchar(50),price float(9,2),description varchar(50));

insert into vehicle values(1,"Activa",80000,"Ride Comfortably"),(2,"Santro",800000,"Safest Car"),(3,"Motor bike",100000,"Ride like Star");

create table vehicle\_history1( vno int primary key,

vname varchar(20), oldprice float(9,2), newprice float(9,2), chdate date,

username varchar(20)

);

create trigger updateVehiclePrice1 before update on vehicle for each row insert into vehicle\_history1(vno,vname,oldprice,newprice,chdate,username) values(vno,old.vname,old.price,new.price,now(),user());

