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( pempno int, pname varchar(20), psal float(9,2), pjob varchar(20), phiredate date )

begin

update emp

set empno=pempno,ename=pname,sal=psal,job=pjob,hiredate=phiredate

where empno=pempno;

end //

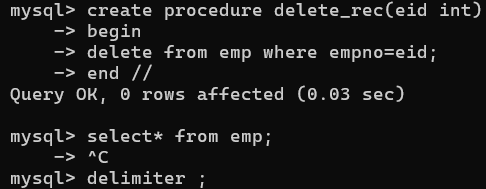
delimiter ;

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

the procedure should accept empno as input parameter.

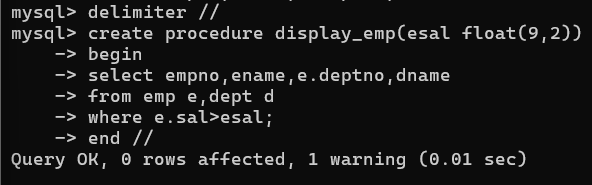
write delete statement inside procedure delete\_emp to delete one record from emp

table



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

> given salary. pass salary as a parameter to procedure



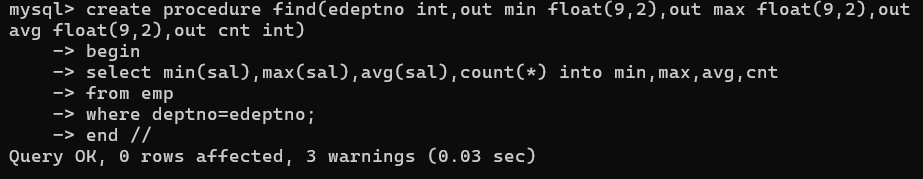
4. write a procedure to find min,max,avg of salary and number of employees in the

given deptno.

deptno --→ in parameter

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

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



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

product,category and salesman table)

create procedure display\_procat(eid int)

begin

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

from product p inner join category c

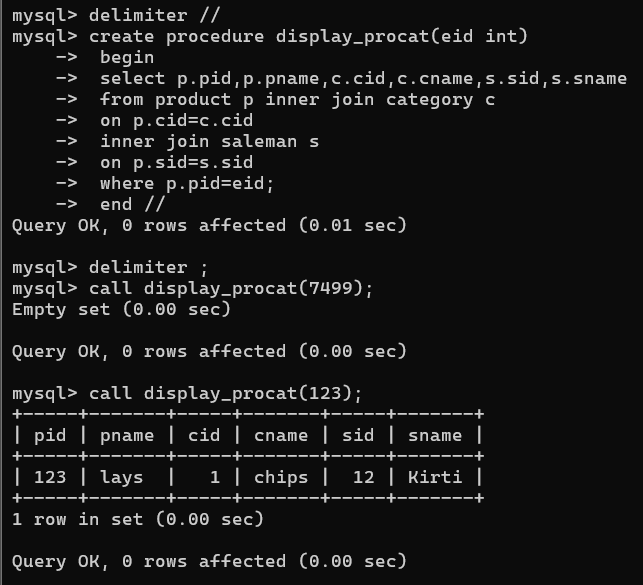
on p.cid=c.cid

inner join saleman s

on p.sid=s.sid

where p.pid=eid;

end //



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)

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 displayemp()

begin

declare vempno,vdeptno,vfinished int default 0;

declare vname,vjob,vstatus 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 vstatus ='greater';

elseif vsal<vavg then

set vstatus='leser';

else

set vstatus='equal';

end if;

select vempno,vname,vjob,vsal,vstatus,vdeptno;

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

delimiter //

mysql> create procedure update\_sal()

-> begin

-> declare esal float(9,2);

-> declare ehiredate date;

-> declare vfinished int default 0;

-> declare exp int default 0;

-> declare empcur cursor for select sal,hiredate from emp;

-> declare continue handler for NOT FOUND set vfinished = 1;

-> open empcur ;

-> label1 : loop

-> fetch empcur into esal,ehiredate;

-> if vfinished = 1 then

-> leave label1;

-> end if;

-> set exp = floor(datediff(curdate(),ehiredate)/365);

-> if exp<=35 then

-> update emp

-> set sal = esal;

-> elseif exp<=38 then

-> update emp

-> set sal=esal\*1.20;

-> else

-> update emp

-> set sal=esal\*1.25;

-> end if;

-> select esal,exp;

-> end loop;

-> close empcur;

-> 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 function calexperience(ehiredate date) returns int

begin

declare vexp int;

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

return vexp;

end//

create procedure empallowance()

begin

declare allowance,vfinished,vempno,experience int default 0;

declare vname varchar(20);

declare vhiredate date;

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

declare continue handler for NOT FOUND set vfinished=1;

open empcur;

label1: loop

fetch empcur into vempno,vname,vhiredate;

if vfinished=1 then

leave label1;

end if;

set experience=calexperience(vhiredate);

set allowance=experience\*3000;

select vempno,vname,vhiredate,experience,allowance;

end loop;

close empcur;

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.

< 1 Year Nil

>=1 Year< 2 Year 10% of Salary

>=2 Year< 4 Year 20% of Salary

>4 Year 30% of Salary

delimiter //

create function companycost(esal float(9,2),ehiredate date) returns float(9,2)

begin

declare da float(9,2);

declare hra float(9,2);

declare ta float(9,2);

declare allowance float(9,2);

declare cost float(9,2);

declare exp int ;

set da=1.15;

set hra=1.20;

set ta=1.08;

set allowance = da\*hra\*ta;

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

if exp<1 then

set cost=esal;

elseif exp<2 then

set cost=esal\*allowance\*1.10;

elseif exp<4 then

set cost=esal\*allowance\*1.20;

else

set cost=esal\*allowance\*1.30;

end if;

return cost;

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)

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(

oldempno int,

newempno int,

oldename varchar(20),

newename varchar(20),

username varchar(20),

changes datetime,

action varchar(20));

create trigger insertemp after insert on emp for each row

insert into emp\_audit(newempno,newename,username,changes,action) values (new.empno,new.ename,user(),now(),'insert');

create trigger deleteemp before delete on emp for each row

insert into emp\_audit(oldempno,oldename,username,changes,action) values (old.empno,old.ename,user(),now(),'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 vehicle\_history(

vno int,

vname varchar(20),

oldprice decimal(9,2),

newprice decimal(9,2),

chdate date,

username varchar(20)

);

create table vehicle\_history(

oldvid int,

oldvname varchar(20),

newvname varchar(20),

oldprice float(9,2),

newprice float(9,2),

olddescription varchar(20),

newdescription varchar(20),

username varchar(20),

changes datetime,

action varchar(20));

create trigger updatevehicle after update on vehicle for each row

insert into vehicle\_history(oldvname,newvname,oldprice,newprice) values(old.vname,new.vname,old.price,new.price);