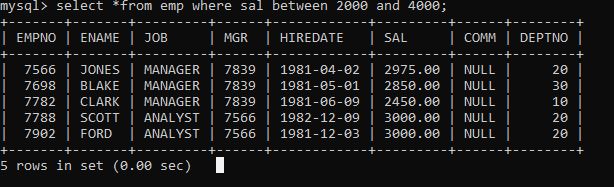
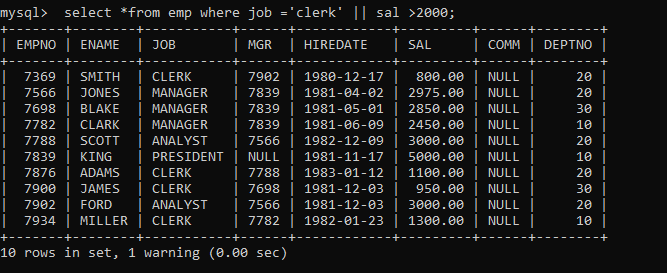
Database Assignment 1

Note : Use Emp, dept and salgrade table

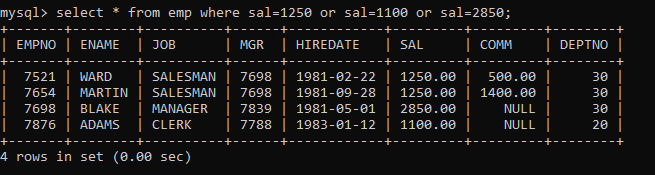
1. To list all records with sal > 2000 and comm>200



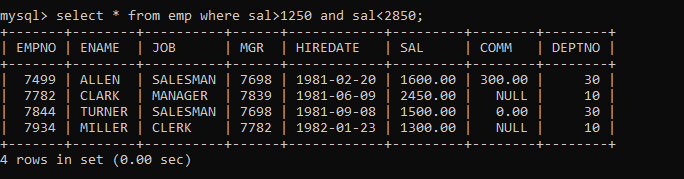
1. To list all record with job=’Clerk’ or sal>2000

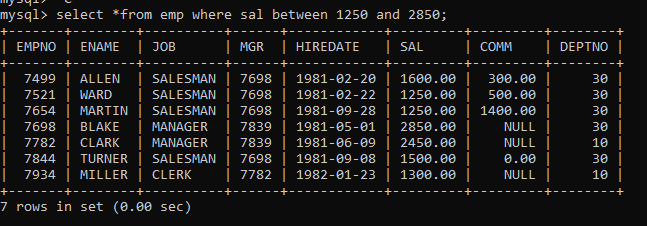


1. To list all the record with sal=1250 or 1100 or 2850

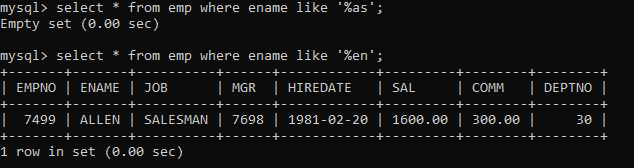


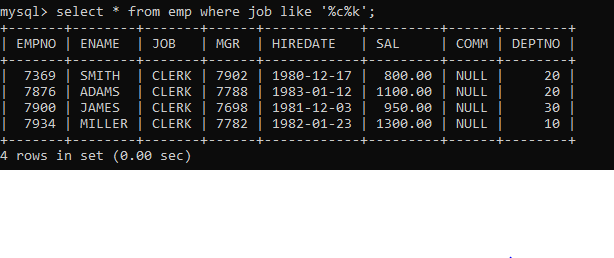
1. To list all employees with sal>1250 and <2850





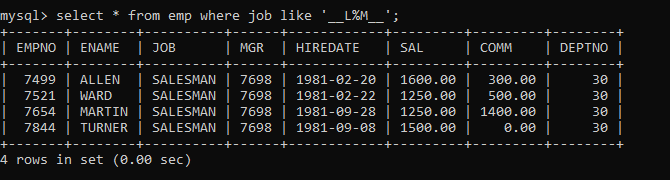
1. To list all employees with name ends with AS



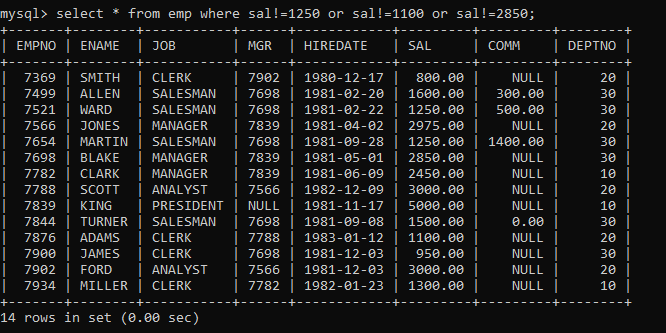
1. To list all employees with job starts with C and

7. To list all employees with job contains L at third position and

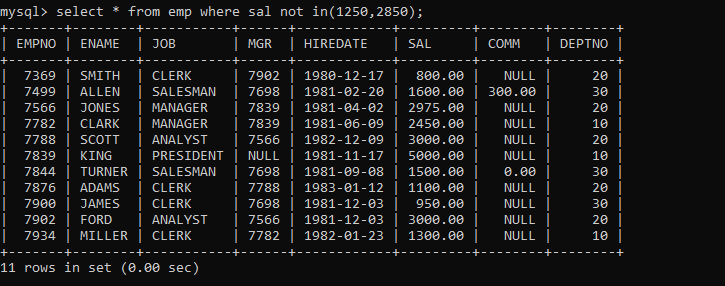
M at third last position



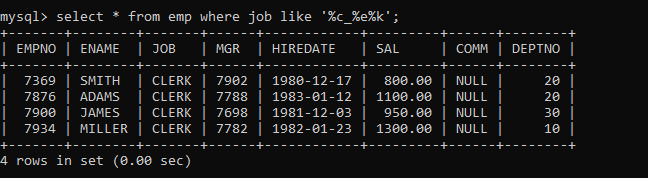
1. To list all the record with sal not equal to 1250 or 1100 or 2850



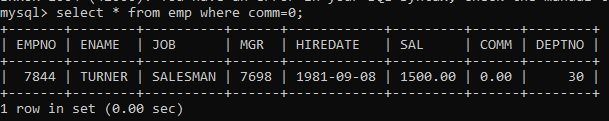
1. To list all employees with salnot >1250 and <2850

,

1. To list all employees with job starts with C , E at 3rd position and ends with K

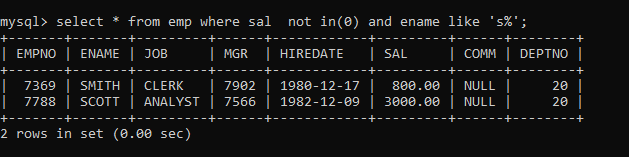


1. To list all rows with comm is null

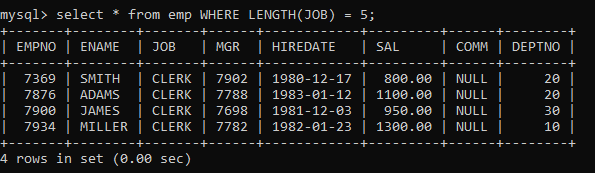


1. To list all employees with sal is null and name starts with ‘S’

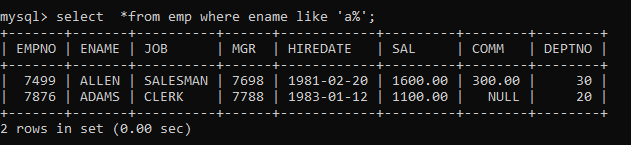
NO ONE WHOES SALARY IS NULL;



1. To list all employees with job contains 5 characters



1. To list all employees with name contain ‘A’ at 1 position and job

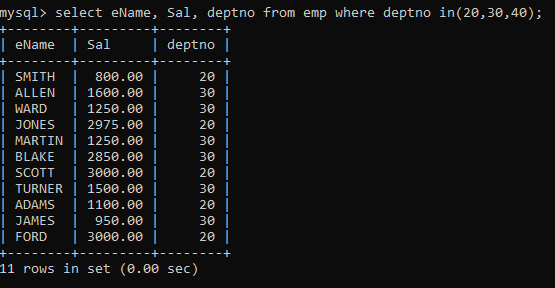


Contains 5 characters

Q2. Solve the following

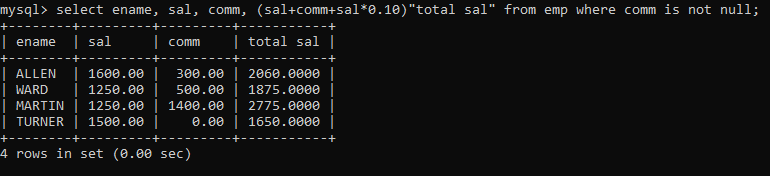
1. Retrieve the details (Name, Salary and dept no) of the emp who are working in

department code 20, 30 and 40.



2. Display the total salary of all employees . Total salary will be calculated as

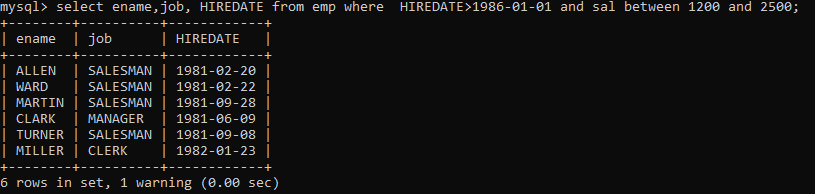
sal+comm+sal\*0.10



3. List the Name and job of the emp who have joined before 1 jan 1986 and whose

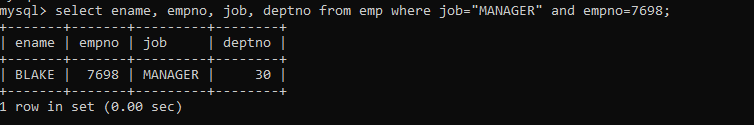
salary range is between 1200and 2500. Display the columns with user defined Column

headers.



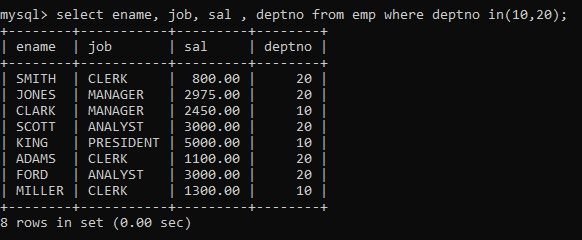
4. List the empno, name, and department number of the emp works under manager

with id 7698



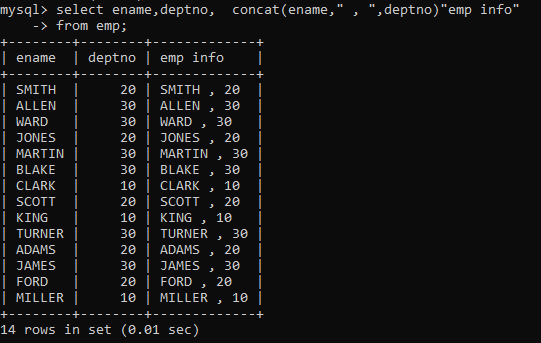
5. List the name, job, and salary of the emp who are working in departments 10 and

30.

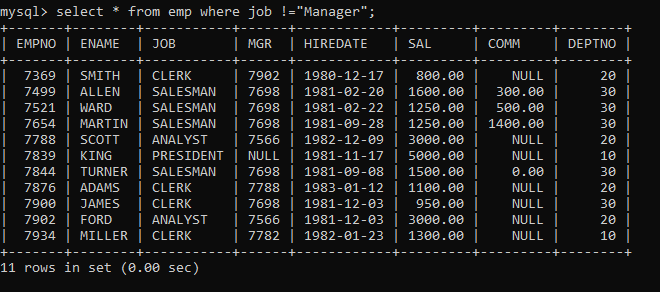


6. Display name concatenated with dept code separated by comma and space. Name

the column as ‘Emp info’.



7. Display the emp details who do not have manager.



8. Write a query which will display name, department no and date of joining of all

employee who were joined January 1, 1981 and March 31, 1983. Sort it based on date of

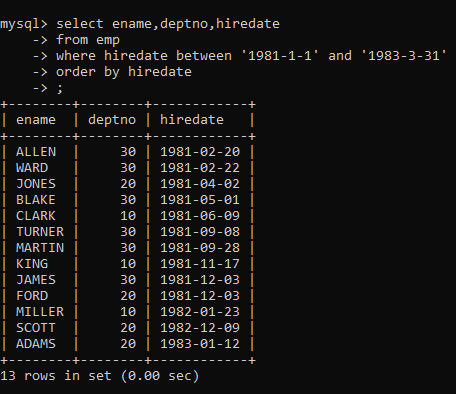
joining (ascending).

select ename,deptno,hiredate

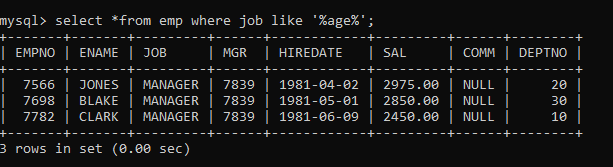
from emp

where hiredate between '1981-1-1' and '1983-3-31'

order by hiredate



9. Display the employee details where the job contains word ‘AGE’ anywhere in the Job



11. List the details of the employee , whose names start with ‘A’ and end with ‘S’ or

whose names contains N as the second or third character, and ending with either ‘N’ or ‘S’.

select \*

from emp

where ename like ‘A%S’ or ename like ‘\_N%N’ or ename like ‘\_N%S’ or ename like ‘\_\_N%N’

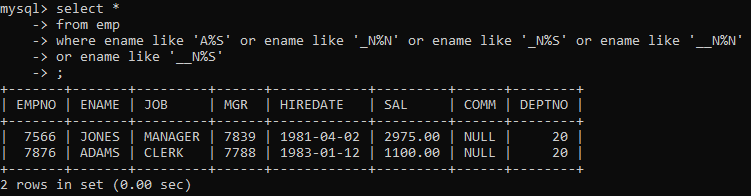
or ename like ‘\_\_N%S’

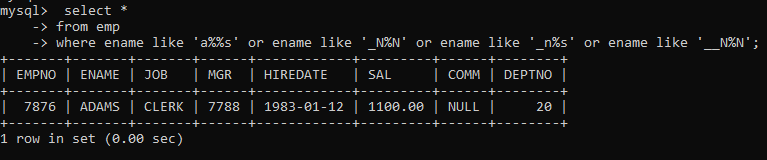
or

select \*

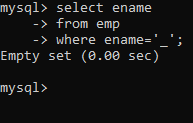
from emp

where ename REGEXP ‘^A.\*S$| ^..?N.\*[NS]$‘





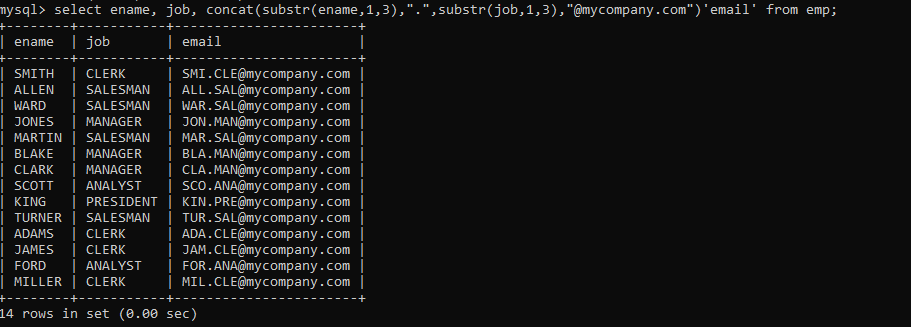
12. List the names of the emp having ‘\_’ character in their name.



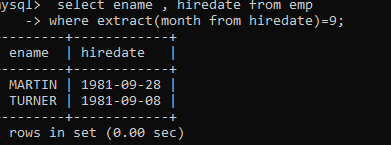
Single Row functions

1. To list all employees and their email, to generate email use 2 to 5 characters from ename

Concat it with 2 to 4 characters in job and then concat it with ‘@mycompany.com’

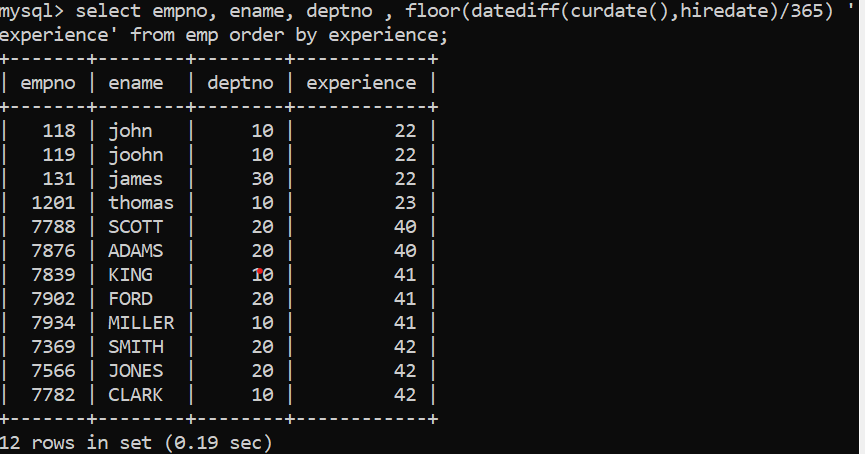


2. List all employees who joined in September.

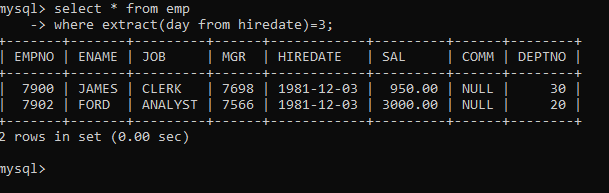


3. List the empno, name, and department number of the emp who have experience of 18 or

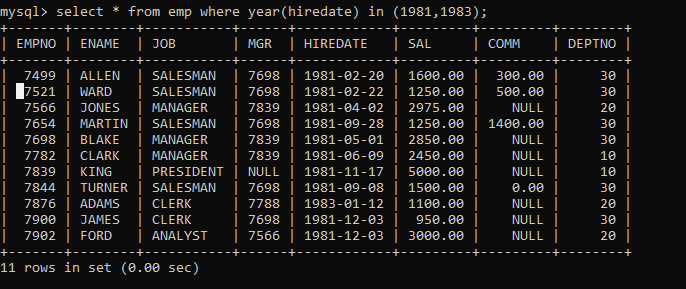
more years and sort them based on their experience.



4. Display the employee details who joined on 3rd of any month or any year



5. display all employees who joined between years 1981 to 1983.

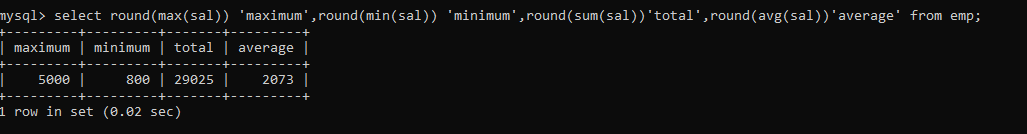


Group functions

6. Display the Highest, Lowest, Total & Average salary of all employee. Label the columns

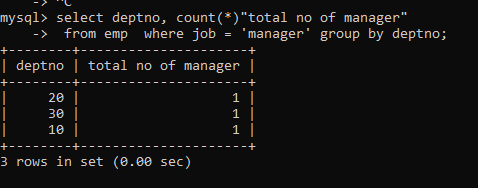
Maximum, Minimum, Total and Average respectively for each Department. Also round the

result to the nearest whole number.



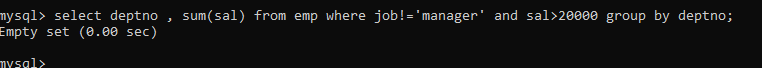
7. Display Department no and number of managers working in that department. Label the

column as ‘Total Number of Managers’ for each department.



8. Get the Department number, and sum of Salary of all non managers where the sum is

greater than 20000.

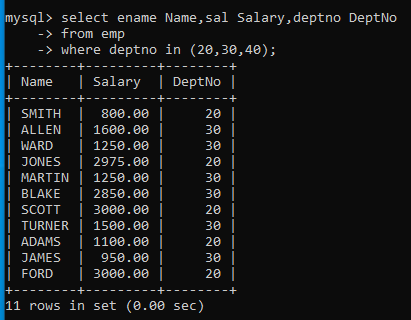


# Mysql assignment 1

**Q2. Solve the following**

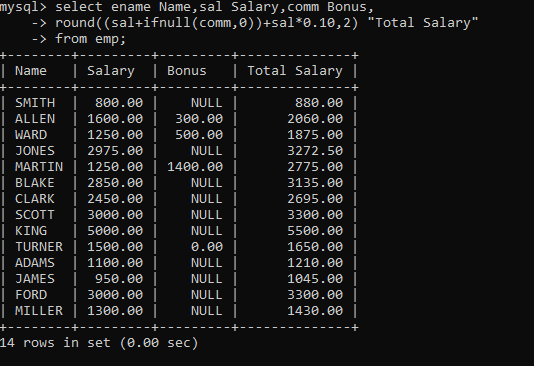
1. Retrieve the details (Name, Salary and dept no) of the emp who are working in

department code 20, 30 and 40.



2. Display the total salary of all employees . Total salary will be calculated as

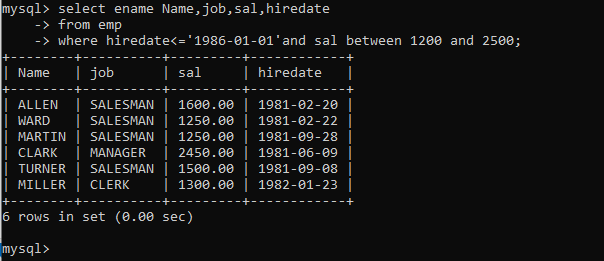
sal+comm+sal\*0.10



3. List the Name and job of the emp who have joined before 1 jan 1986 and whose

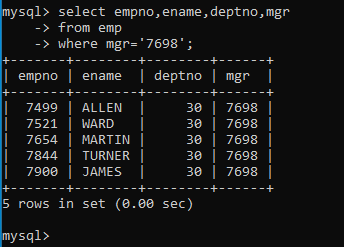
salary range is between 1200and 2500. Display the columns with user defined Column

headers.



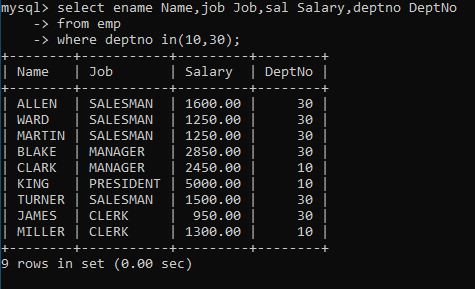
4. List the empno, name, and department number of the emp works under manager

with id 7698



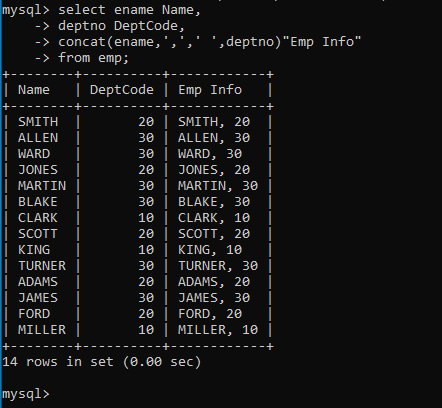
5. List the name, job, and salary of the emp who are working in departments 10 and

30.

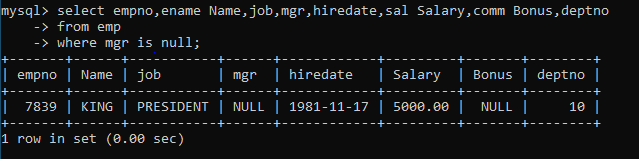


6. Display name concatenated with dept code separated by comma and space. Name

the column as ‘Emp info’.



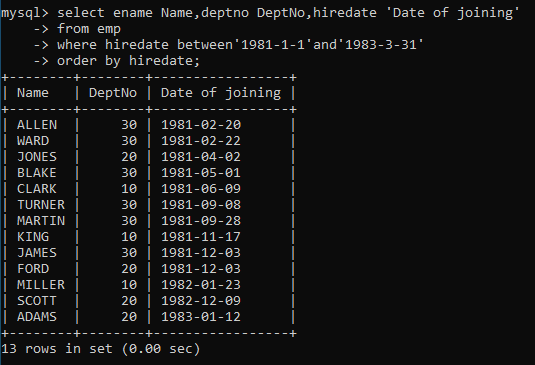
7. Display the emp details who do not have manager.



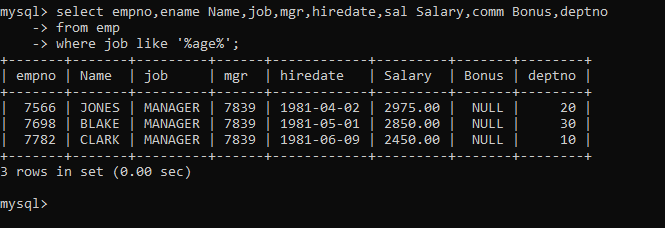
8. Write a query which will display name, department no and date of joining of all

employee who were joined January 1, 1981 and March 31, 1983. Sort it based on date of

joining (ascending).

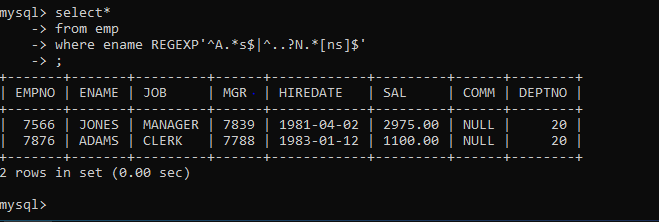


9. Display the employee details where the job contains word ‘AGE’ anywhere in the Job

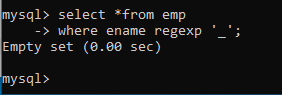


11. List the details of the employee , whose names start with ‘A’ and end with ‘S’ or

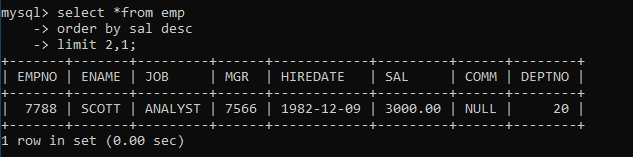
whose names contains N as the second or third character, and ending with either ‘N’ or ‘S’.



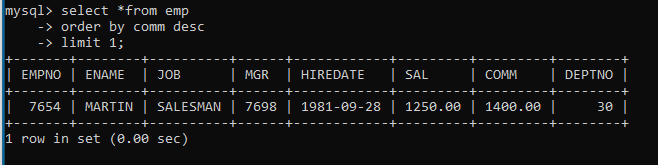
12. List the names of the emp having ‘\_’ character in their name.



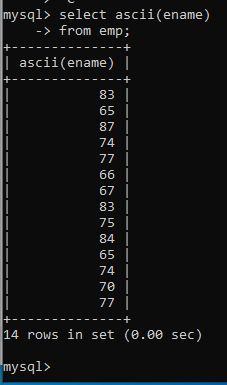
13. find 3rd highly paid employees



14. find employee who has earned highest commission

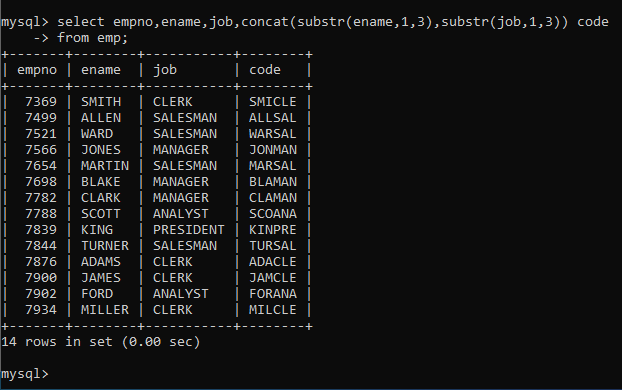


15. display ascii value of 1st character of job from emp



16. display empno,ename,job,code. code should be 1 st 3 characters of ename and 1

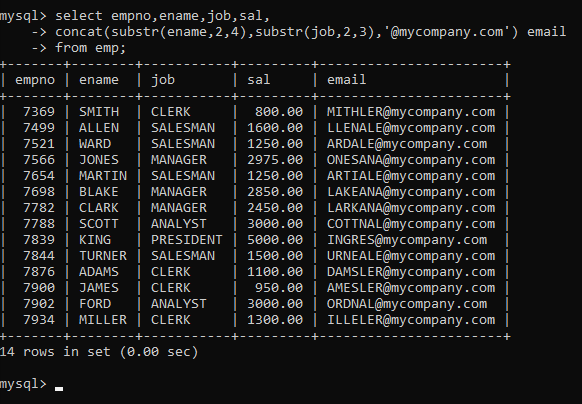
st 3 characters of job



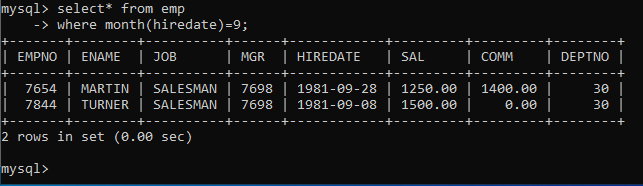
Single Row functions

1. To list all employees and their email, to generate email use 2 to 5 characters from ename

Concat it with 2 to 4 characters in job and then concat it with ‘@mycompany.com’



2. List all employees who joined in September.



3. List the empno, name, and department number of the emp who have experience of 18 or

more years and sort them based on their experience.

4. Display the employee details who joined on 3rd of any month or any year

5. display all employees who joined between years 1981 to 1983.

**Day 3**

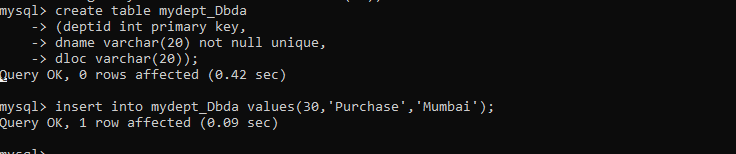
Q1:-create table mydept\_DBDA (

deptid number primary key,

dname varchar2(20) not null unique, dloc varchar2(20)

)

insert into mydept\_DBDA values(30,'Purchase','Mumbai');



Q2. create table myemployee (

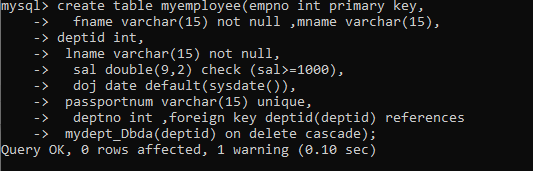
empno number(5) primary key,fname varchar2(15) not null, mname varchar2(15),

lname varchar2(15) not null,

sal number(9,2) check(sal >=1000), doj date default sysdate, passportnum varchar2(15) unique,

deptno number constraint fk\_deptno references mydept\_DBDA(deptid) on delete cascade

)



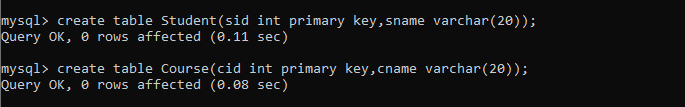
Q3 .Create following tables Student, Course

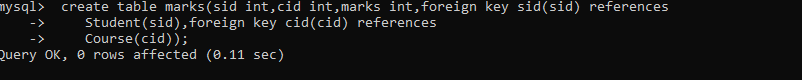
Student (sid,sname) ---------------- sid ---primary key

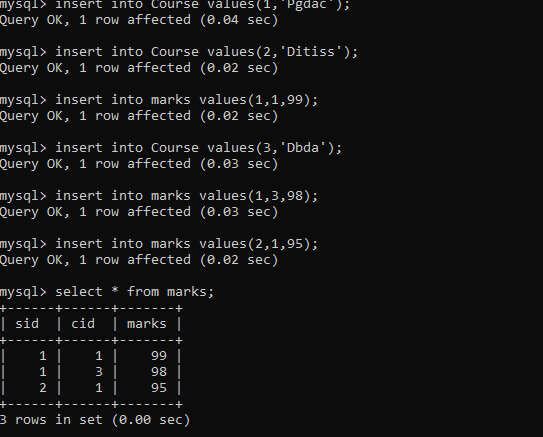
Course(cid,cname)-------------- cid ---primary key

Marks(studid,coursed,marks) Sample data for marks table studid,courseid,marks

|  |  |  |
| --- | --- | --- |
| 1 | 1 | 99 |
| 1 | 3 | 98 |
| 2 | 1 | 95 |
| 2 | 2 | 97 |





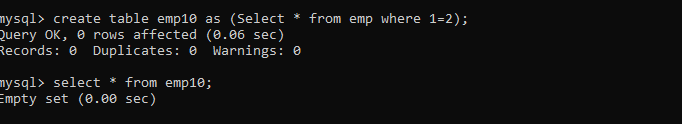


Q4. Create empty table emp10 with table structure same as emp table. create table emp10 as

(

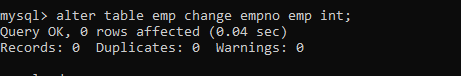
select \* from emp where 1=2;

)

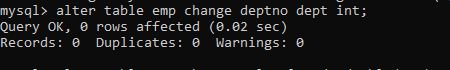


Q5. Solve following using alter table

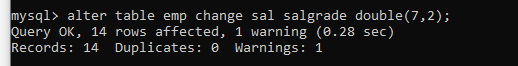
add primary key constraint on emp,dept,salgrade emp ---- empno



dept--- deptno



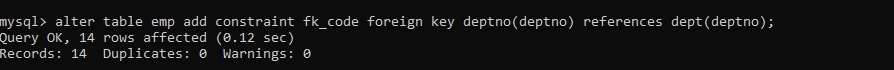
salgrade--- grade



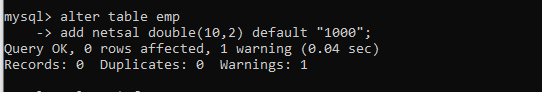
add foreign key constarint in emp

deptno --->> dept(deptno)



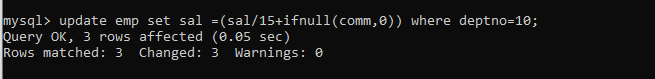


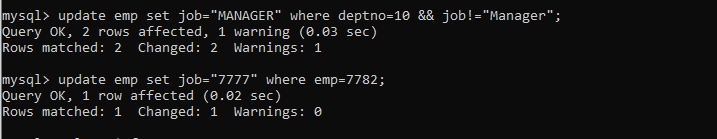
add new column in emp table netsal with constraint default 1000

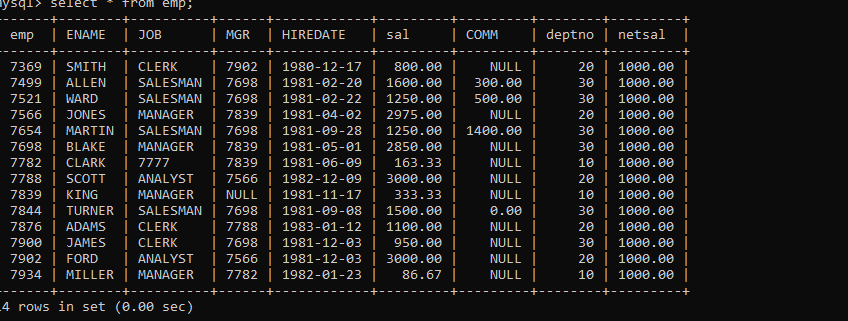


Q6.Update employee sal increase sal of each employee by 15 % sal +comm, change the job

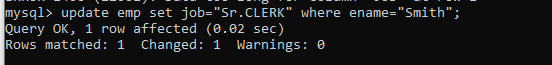
to manager and mgr to 7777 for all employees in deptno 10.



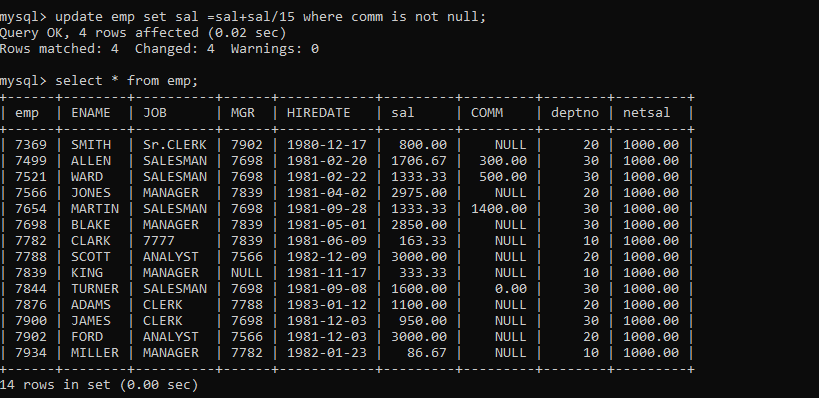




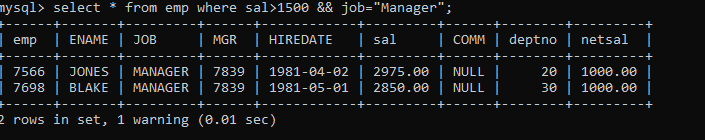
Q7.change job of smith to senior clerk



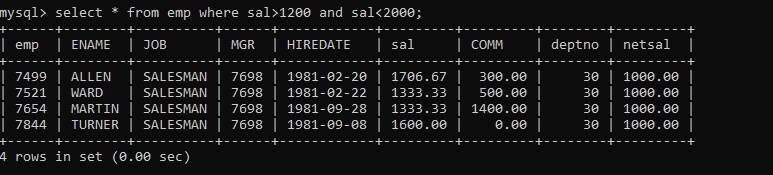
Q8. increase salary of all employees by 15% if they are earning some commission



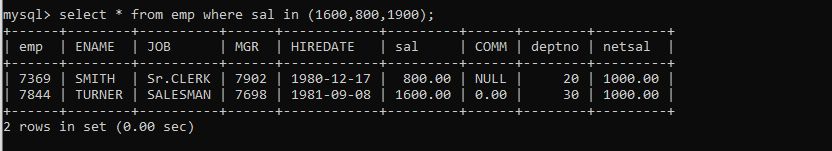
10. To find all managers with salary >1500



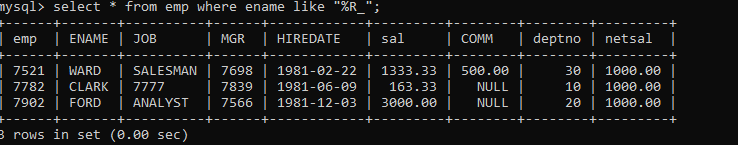
11.list all employees with sal >1200 and < 2000



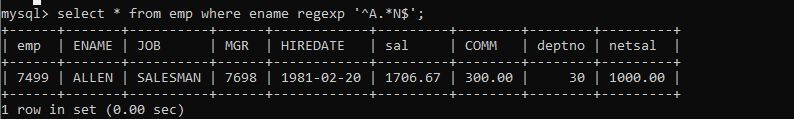
12. list all employees with sal is 1600 or sal is 800 or sal is 1900



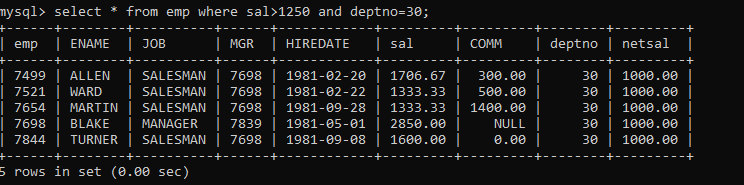
1. 13. list all employees with R at second last position in name



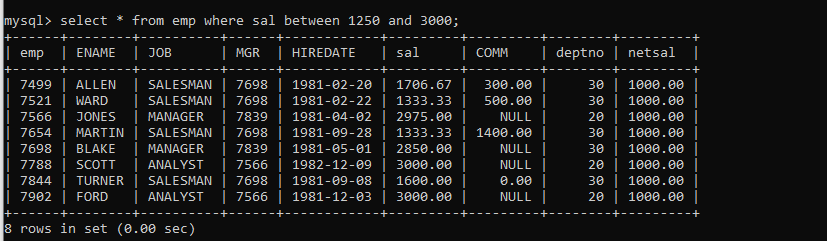
14. List all employees with name starts with A and ends with N



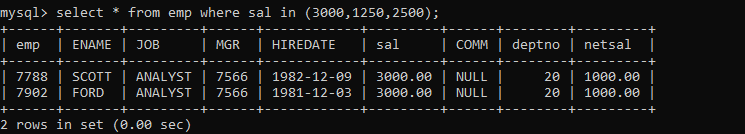
15. list all employees with salary > 1250 and dept no=30



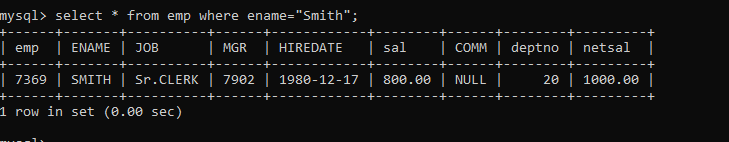
16.list all employees with salary >=1250 and <= 3000



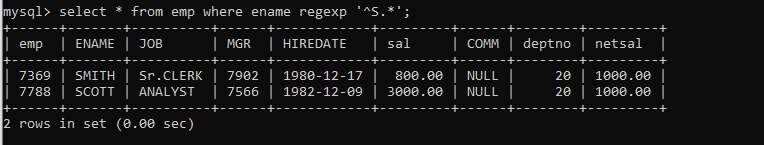
17. list all employees with salary either equal to 3000 or 1250 or 2500



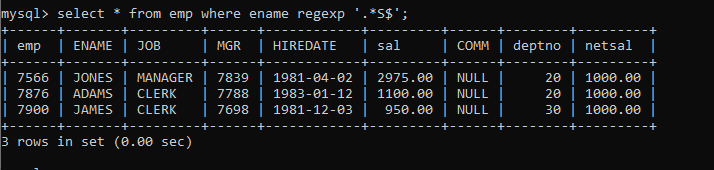
18. list all employee with name=SMITH



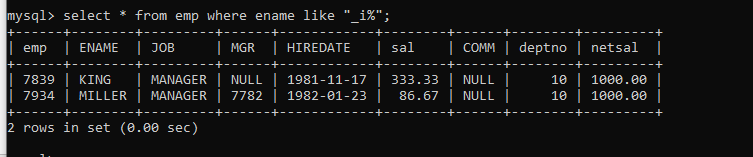
19. **list all employees with name starting with S**



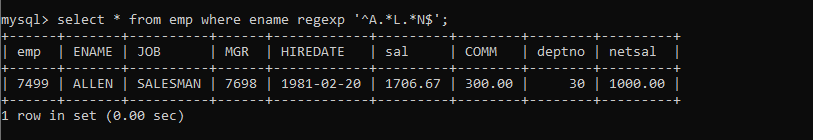
20. list all employees with name ending with S



21. list all employees with name contains I at 2nd position



22.list all employees with name starts with A ends witn N and somewhere in between L is there

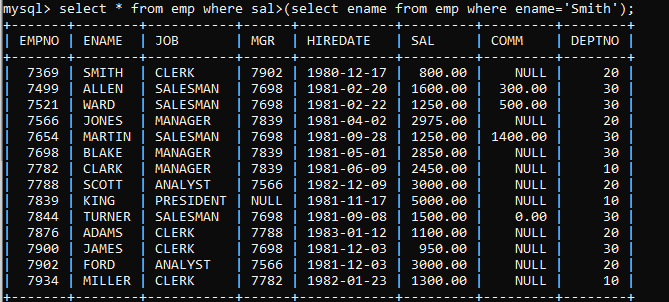


23. list all employees with name starts with A and B at 3 rd position and P at second last position

==================================Day5============================================

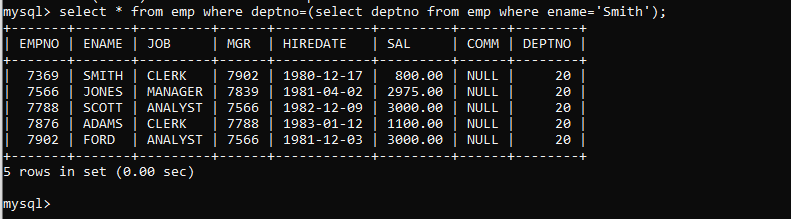
38. list all employees with sal>smith's sal

select \* from emp where sal>(select ename from emp where ename='Smith');



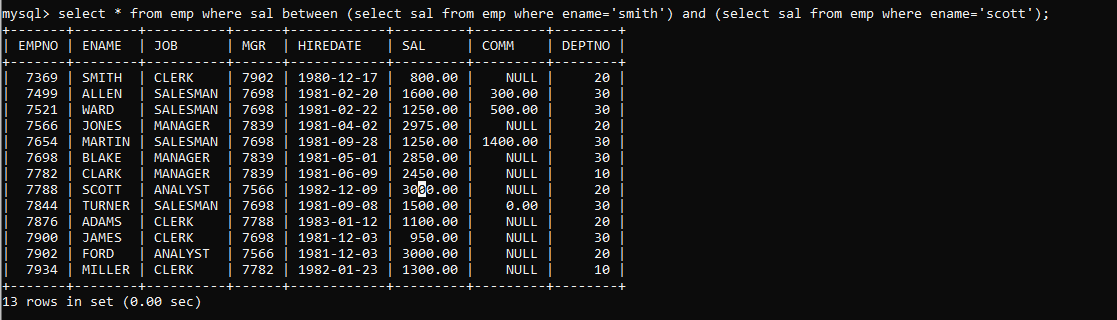
39. . list all employees who are working in smith's department

select \* from emp where deptno=(select deptno from emp where ename='Smith');



40. list all employees with sal < rajan's sal and salary > revati's sal

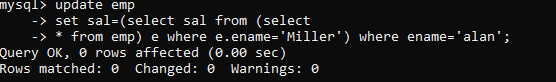
select \* from emp where sal between (select sal from emp where ename='smith') and (select sal from emp where ename='scott');



Q 41.delete from emp where dept is of allen



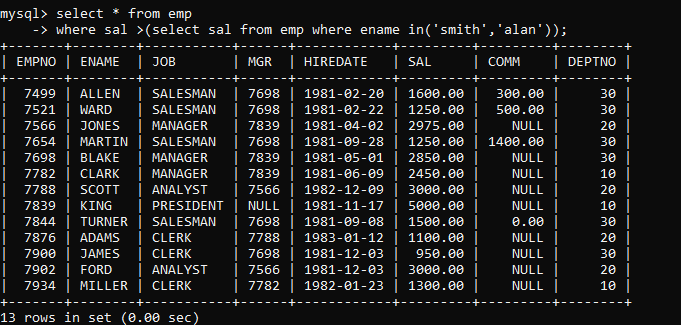
Q 42 allen sal to miller change



44. list all employees with salary > either Smith's salary or alan's sal

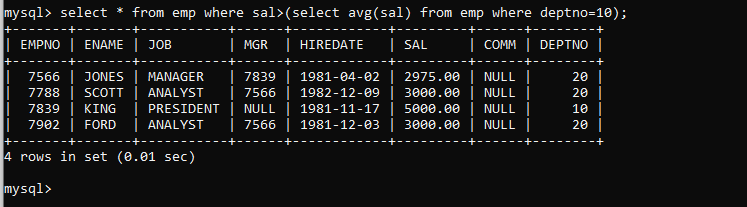
select \* from emp

-> where sal >(select sal from emp where ename in('smith','alan'));



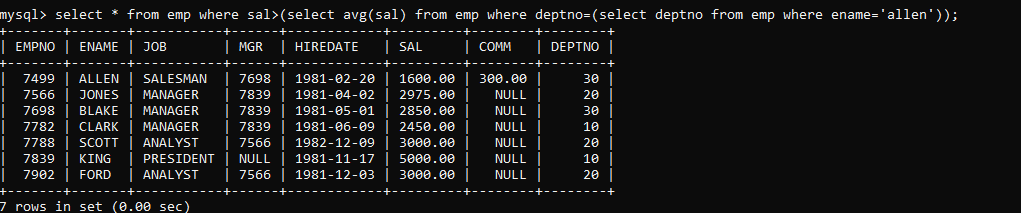
45. list all employees who earn more than average sal of dept 10

select \* from emp where sal>(select avg(sal) from emp where deptno=10);



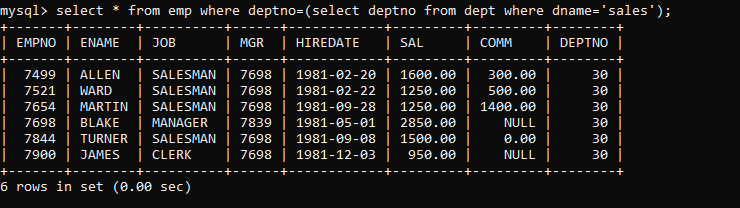
46, list all employees who earn more than average sal of Alan's department

select \* from emp where sal>(select avg(sal) from emp where deptno=(select ename from emp where ename='alan'));

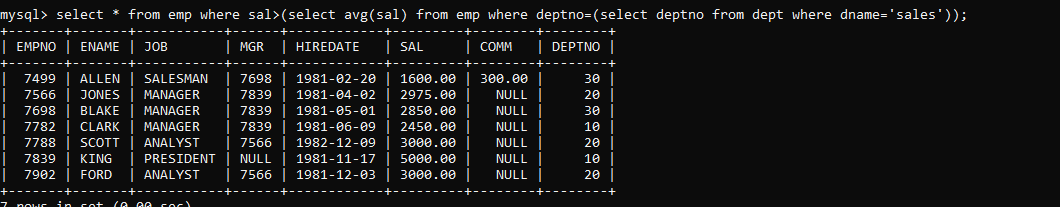


47. list all employees who are working in purchase department

select \* from emp where deptno=(select deptno from dept where dname='sales');



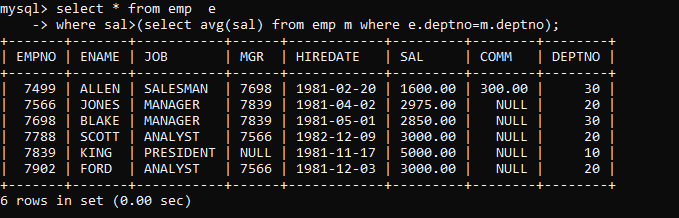
Self



48. list all employees who earn more than average salary of their own department

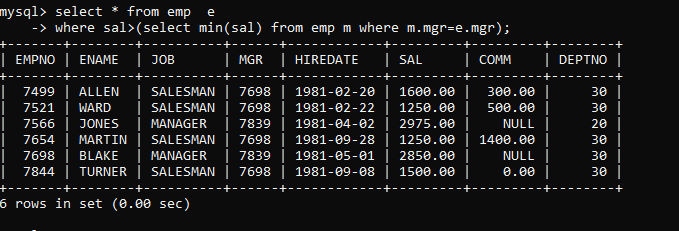
select \* from emp e

-> where sal>(select avg(sal) from emp m where e.deptno=m.deptno);



49.list all employees who earn sal < than their managers salary

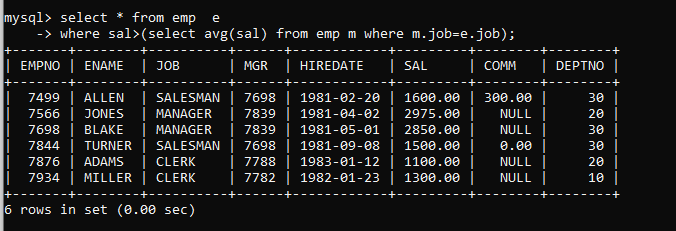
select \* from emp e where sal>(select sal from emp m where m.mgr=e.mgr);



50. list all employees who are earning more than average salary of their job

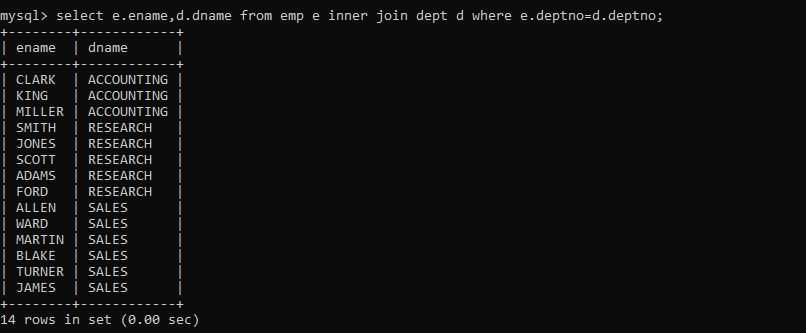
select \* from emp e

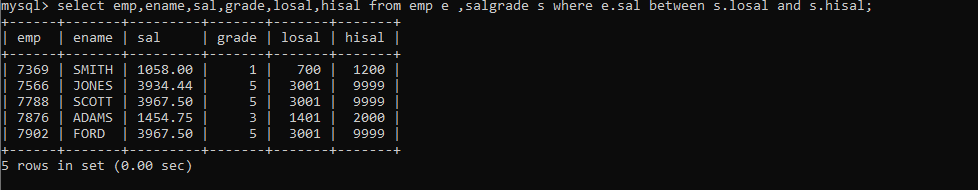
-> where sal>(select avg(sal) from emp m where m.job=e.job);



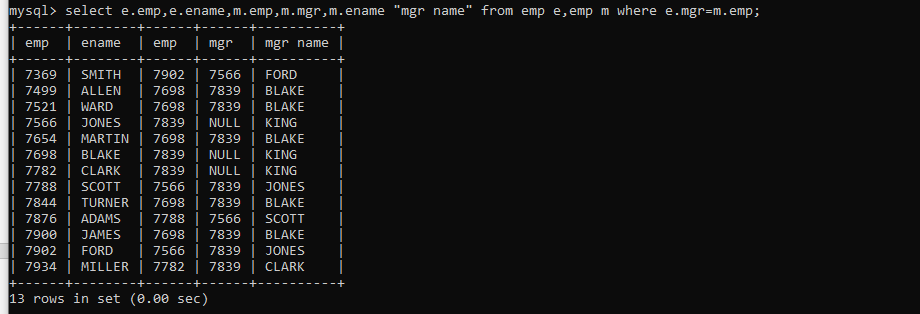
51.display employee name and department

51.display employee name and department

52.display empno,name,department name and grade (use emp,dept and salgrade table)



53.list all employees number,name, mgrno and manager name



54. 4. create following tables and solve following questions(primary keys are marked in yellow) foreign keys are marked in green product(pid,pname,price,qty,cid,sid) salesman (sid,sname,address) category(cid,cnam,descritpion) 1. list all product name,their category name and name of a person, who sold that product 2. list all product name and salesman name for all salesman who stays in pune 3. list all product name and category name

create table salesman(

sid int primary key, sname varchar(25) not null, address varchar(70) default "pune");

create table category(

cid int primary key, cname varchar(25) not null, descr varchar(70) default "This is a product");

create table product (

pid int primary key,

pname varchar(50),

price int,

qty int,

cid int,

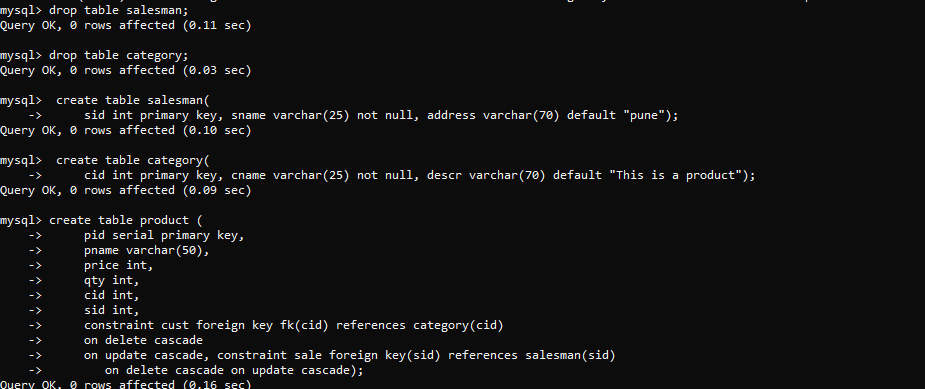
sid int,

constraint cust foreign key fk(cid) references category(cid)

on delete cascade

on update cascade, constraint sale foreign key(sid) references salesman(sid)

on delete cascade on update cascade);



55.

create table faculty(

fid int primary key,

fname varchar(20) not null,

sp\_skill1 varchar(20),

sp\_skill2 varchar(20)

);

create table room(

roomid int primary key,

rname varchar(20) not null,

rloc varchar(20)

);

create table course1(

cid int primary key,

cname varchar(20),

rid int,

fid int,

constraint fk\_rid foreign key(rid)

references room(roomid)

on update cascade

on delete set null,

constraint fk\_fid foreign key(fid)

references faculty(fid)

on update cascade

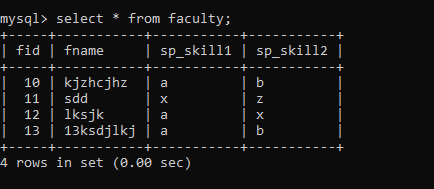
on delete set null

);

insert into faculty values(10,'kjzhcjhz','a','b'),

(11,'sdd','x','z'),(12,'lksjk','a','x'),

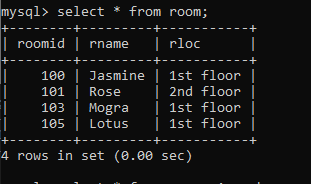
(13,'13 ksdjlkj','a','b');



insert into room values(100,'Jasmine','1st floor'),

(101,'Rose','2nd floor'),(105,'Lotus','1st floor'),

(103,'Mogra','1st floor');

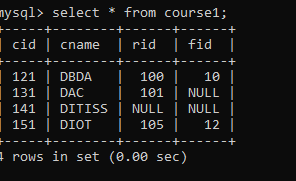


insert into course1 values(121,'DBDA',100,10);

insert into course1(cid,cname,rid) values(131,'DAC',101);

insert into course1(cid,cname) values(141,'DITISS');

insert into course1 values(151,'DIOT',105,12);



55.1 list all courses for which no room is assigned. And all rooms for which are available

select cid,cname,rid,null roomid,null rname,null rloc

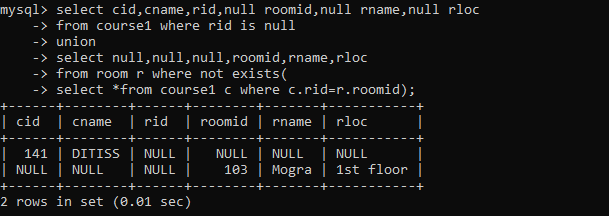
from course1 where rid is null

union

select null,null,null,roomid,rname,rloc

from room r where not exists(

select \*from course1 c where c.rid=r.roomid);



55.2 list all faculties who are not allocated to any course and rooms which are not allocated to any course

select f.fid,fname,null roomid,null rname,null rloc

from faculty f where not exists

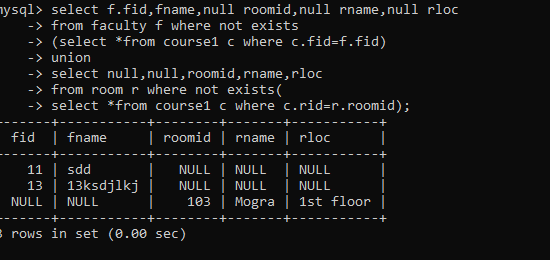
(select \*from course1 c where c.fid=f.fid)

union

select null,null,roomid,rname,rloc

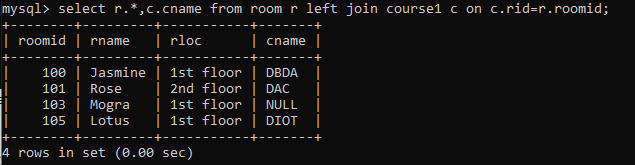
from room r where not exists(

select \*from course1 c where c.rid=r.roomid);

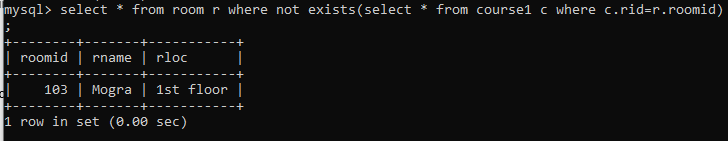


55.3 list all rooms which are allocated or not allocated to any courses

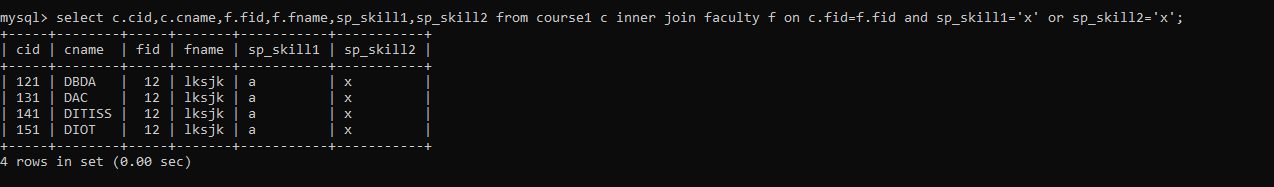
select r.\*,c.cname from room r left join course1 c on c.rid=r.roomid;



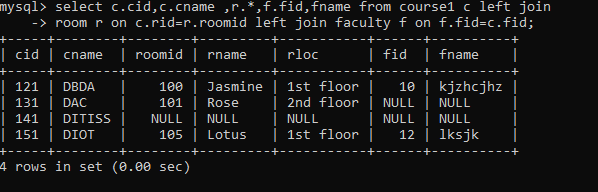
55.4 list all rooms which are not allocated to any courses



* 1. display courses and faculty assigned to those courses whose special skill is database



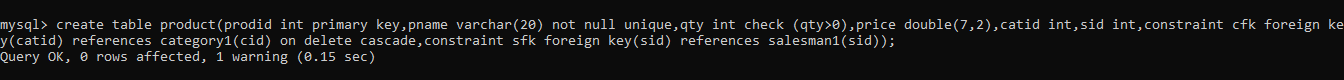
55.6 display time table --- it should contain course details , faculty and room details

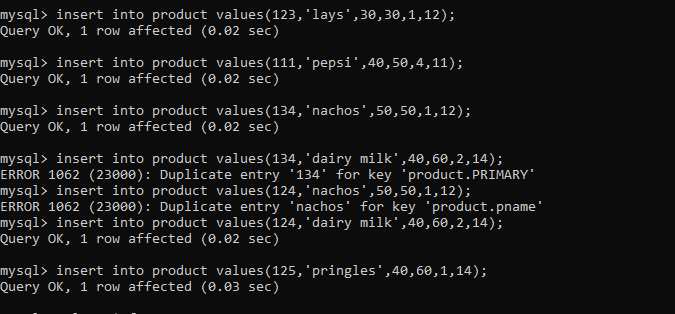


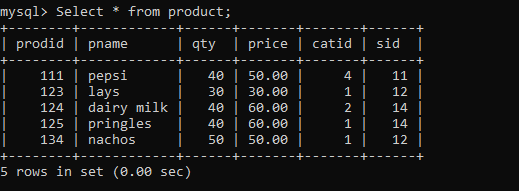
56.create following tables with given constraints product---- qty >0, default 20.00,pname not null and unique

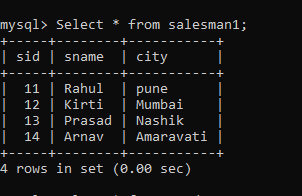
prodid pname qty price catid sid

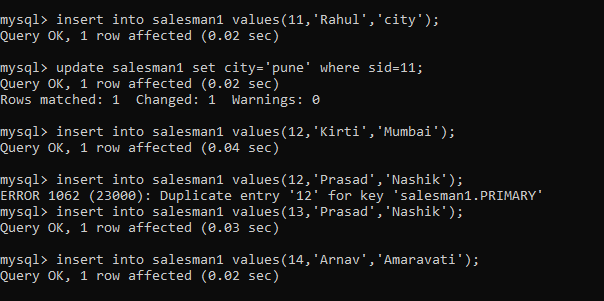
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 123 | lays | 30 | 30.00 | 1 | 12 |
| 111 | pepsi | 40 | 50.00 | 4 | 11 |
| 134 | nachos | 50 | 50.00 | 1 | 12 |
| 124 | dairy milk | 40 | 60.00 | 2 | 14 |
| 124 | pringles 40 | 60.00 | 1 14 |  |  |

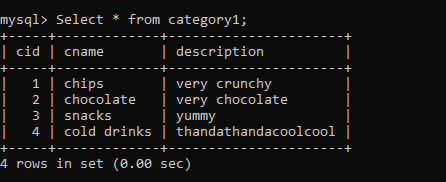


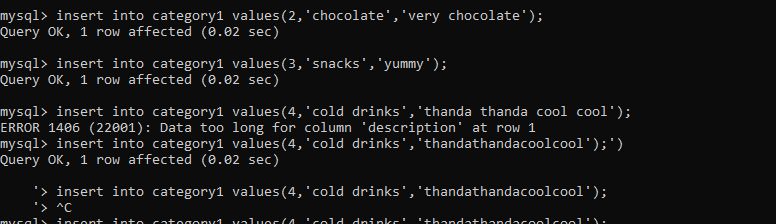




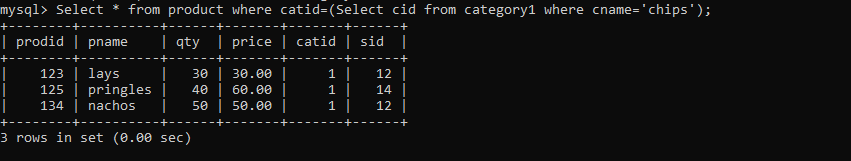




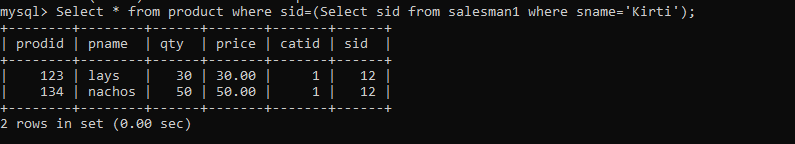




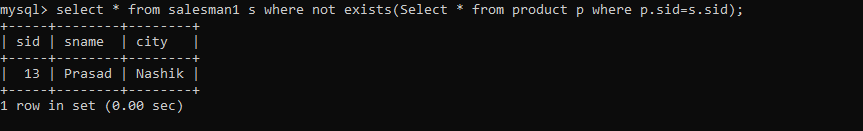
1. List all products with category chips



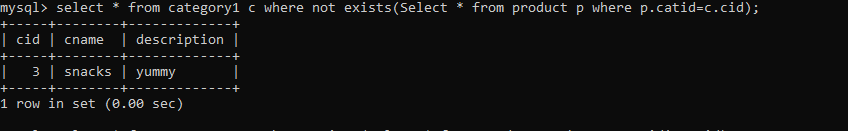
2.display all products sold by kirti



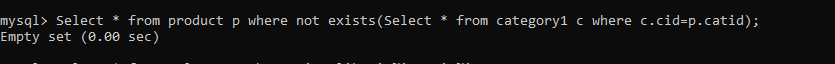
1. display all salesman who do not sold any product



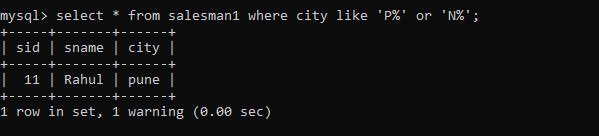
4.display all category for which no product is there



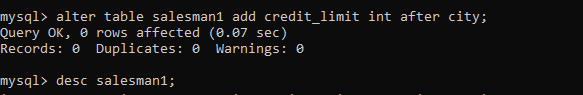
5.display all products with no category assigned



6.list all salesman who stays in city with name starts with P or N



7.add new column in salesman table by name credit limit

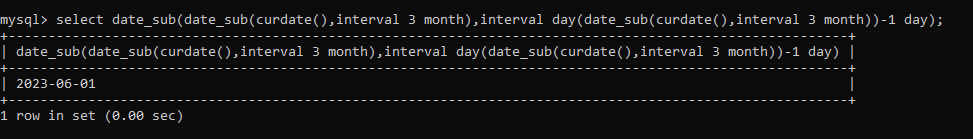


# MYSQL-date function assignment

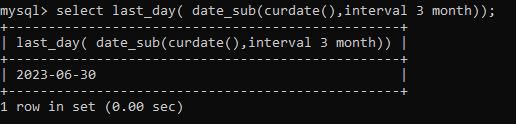
1.Write a query to display the first day of the month (in datetime format) three months before the current month.

Sample current date : 2014-09-03

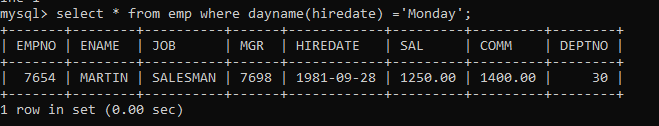
Expected result : 2014-06-01



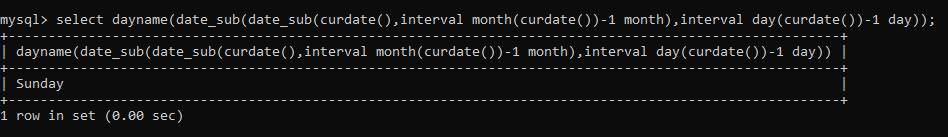
2. Write a query to display the last day of the month (in datetime format) three months before the current month.



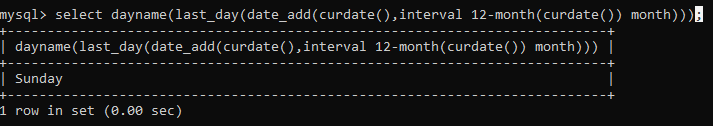
3. Write a query to get the distinct Mondays from hiredate in emp tables.



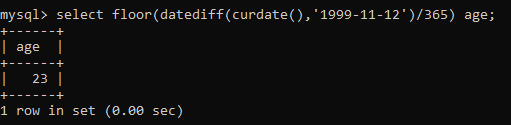
4. Write a query to get the first day of the current year.



5. Write a query to get the last day of the current year.



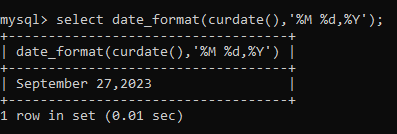
6. Write a query to calculate your age in year.



7. Write a query to get the current date in the following format.

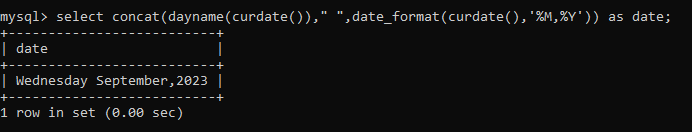
Sample date : 04-sep-2014

Output : September 4, 2014

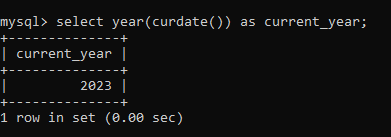


8. Write a query to get the current date in Thursday September 2014 format.

Thursday September, 2014



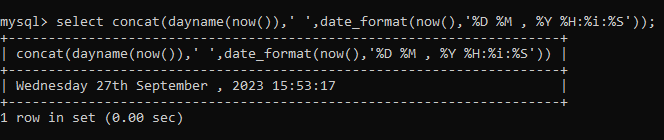
9. Write a query to extract the year from the current date.



10. Write a query to get the first name and hire date from employees table where hire date between '1987-06-01' and '1987-07-30'

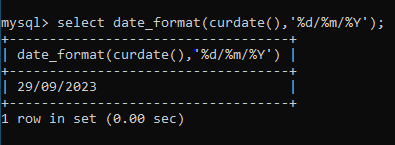
11. Write a query to display the current date in the following format.

Sample output: Thursday 4th September 2014 00:00:00



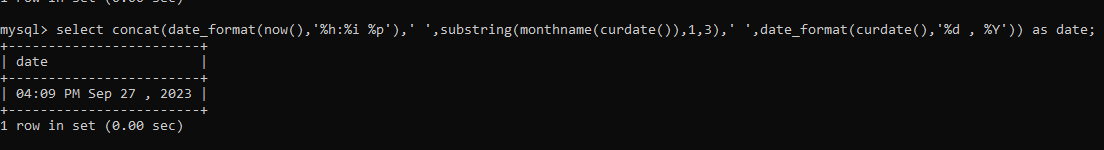
12. Write a query to display the current date in the following format.

Sample output: 05/09/2014

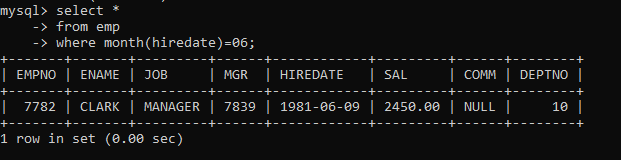


13. Write a query to display the current date in the following format.

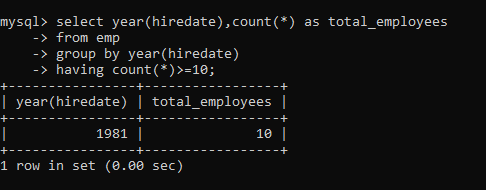
Sample output: 12:00 AM Sep 5, 2014



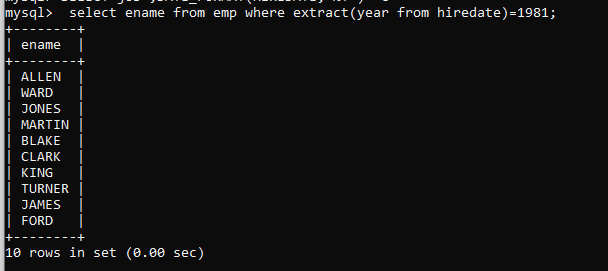
14. Write a query to get the employees who joined in the month of June.

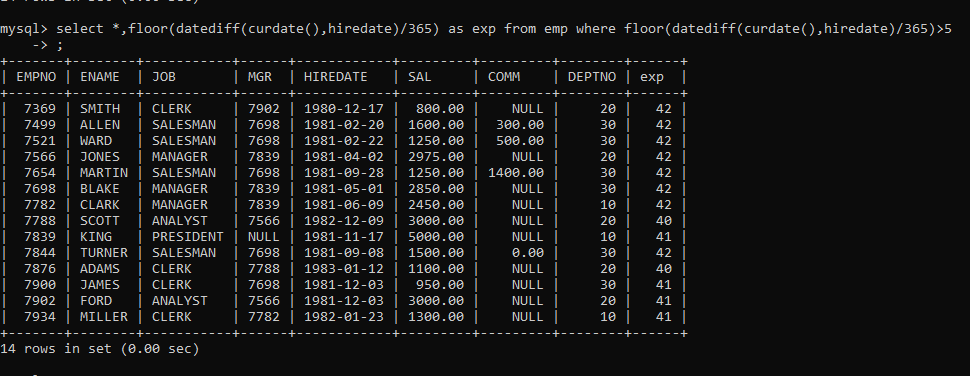


15. Write a query to get the years in which more than 10 employees joined.

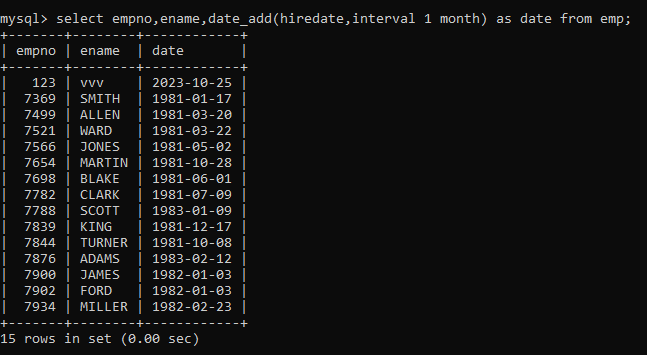


16. Write a query to get first name of employees who joined in 1987.

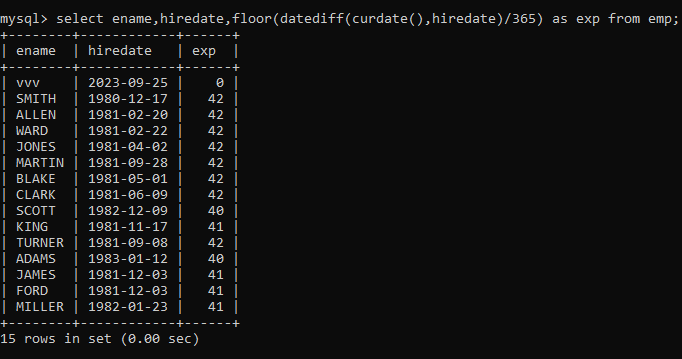
****

17. Write a query to get employees whose 

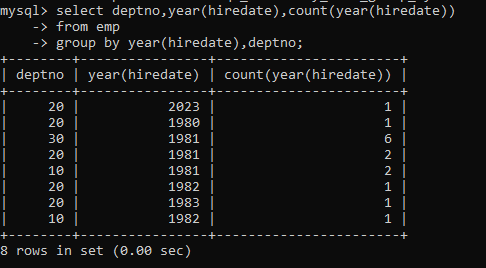
18. Write a query to get employee ID, last name, and date of first salary of the employees.

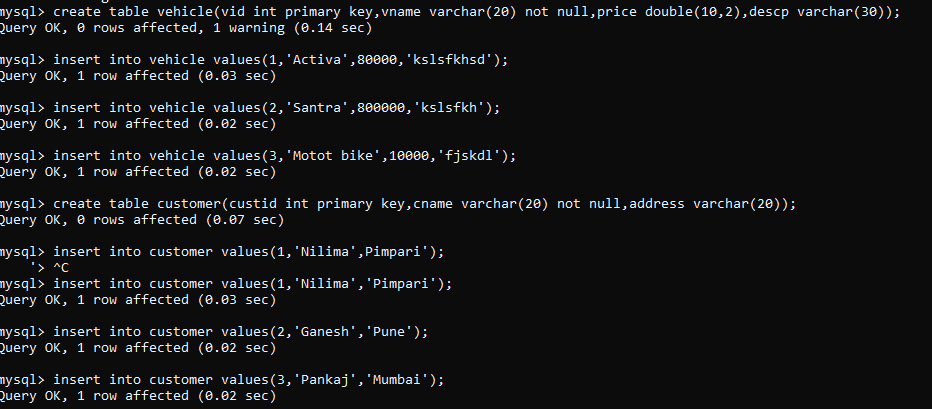


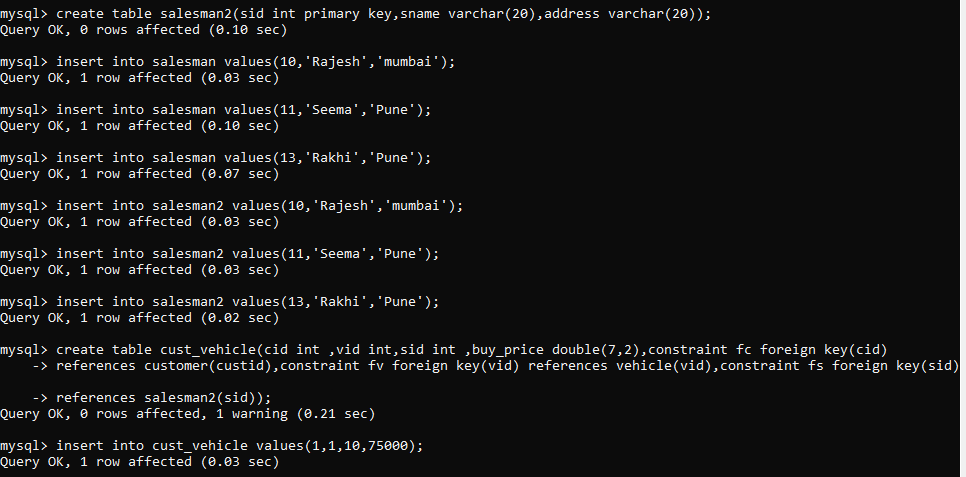
19. Write a query to get first name, hire date and experience of the employees.

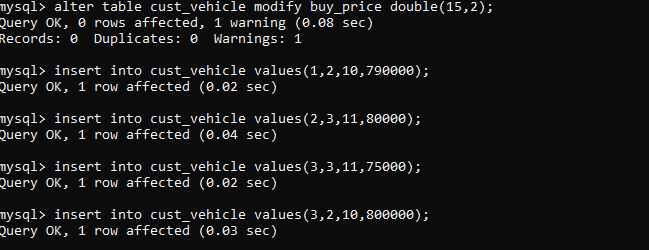


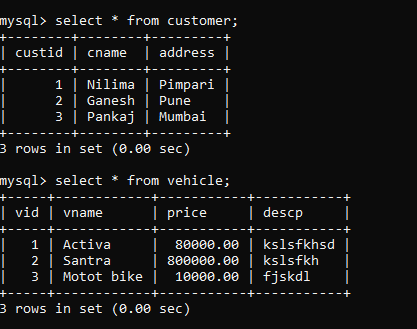
20. Write a query to get the department ID, year, and number of employees joined.

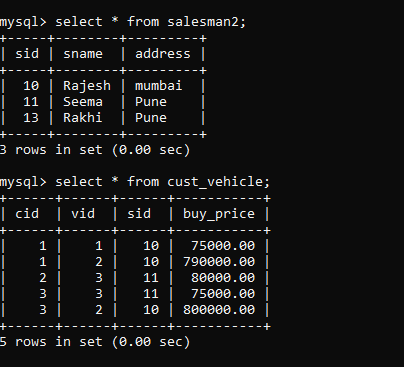




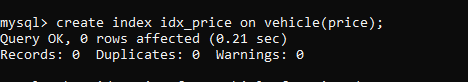


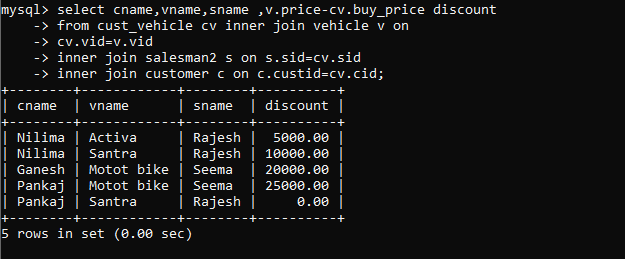




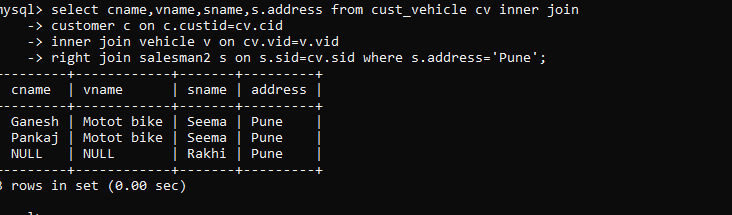


2.create index on vehicle based on price

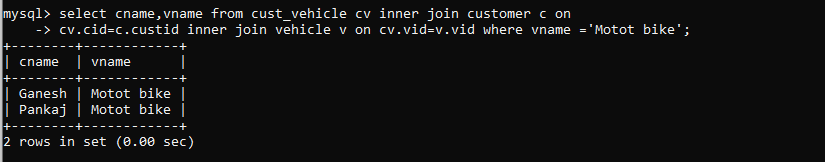


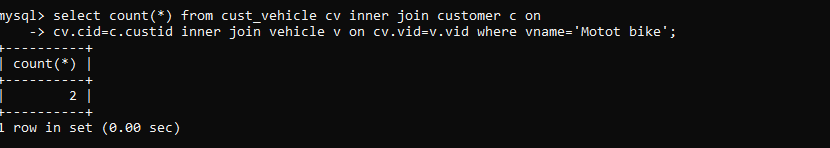
1. find all customer name vehicle name salesman2 name discount earn by all customer
2. 

4. find all customer name,vehicle name,salesman name for all salesman who stays in pune



5.find how many customers bought motor bike





6.create a view find\_discount which displays output

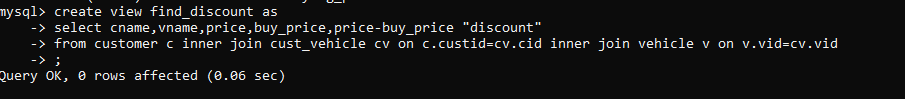
-------to create view create view find\_discount as

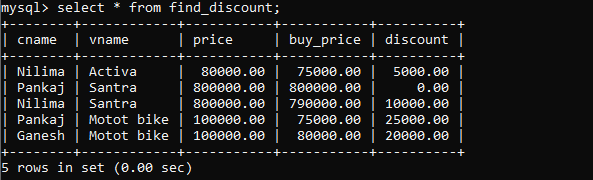
select cname,vname,price,buying\_price,price-buying\_price “discount”

from customer c inner join cust\_vehicle cv on c.custid=cv.cid inner join vehicle v on v.vid=cv.vid

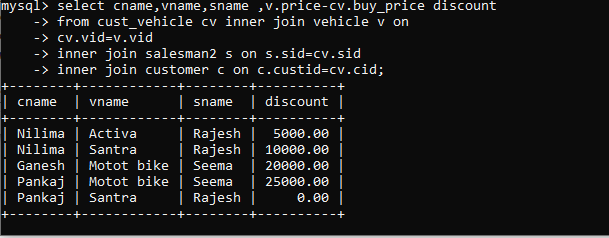
--------to display discount

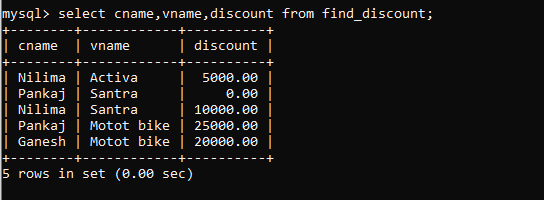
select \* from find\_discount;



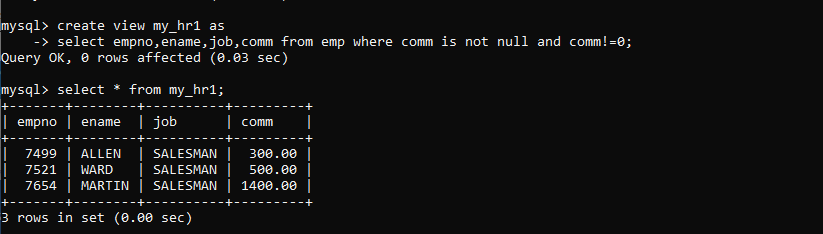


7.find all customer name, vehicle name, salesman name, discount earn by all customer

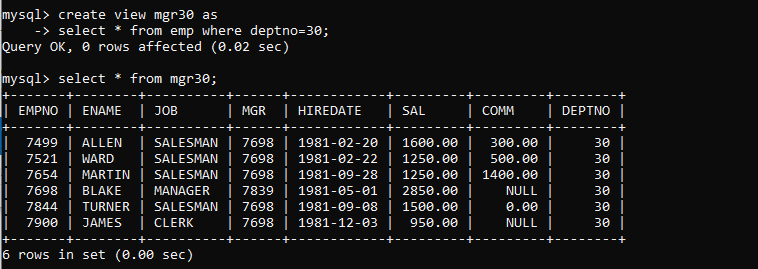




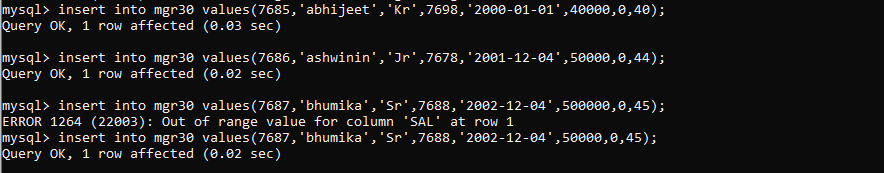
8.create view my\_hr to display empno,ename,job,comm for all employees who earn commission

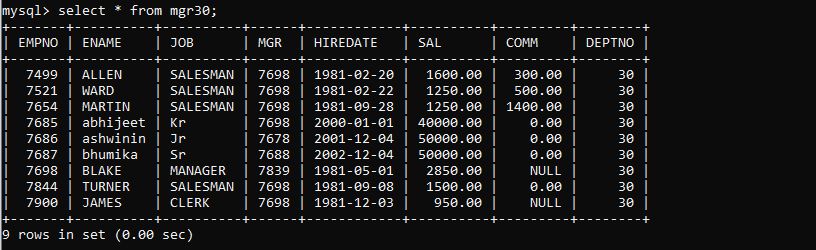


1. create view mgr30 to display all employees from department 30

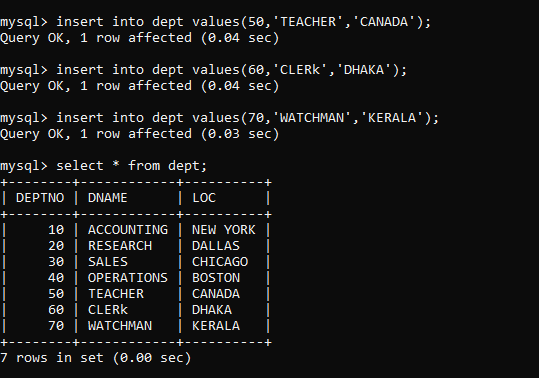


10. insert 3 employees in view mgr30 check whether insertion is possible



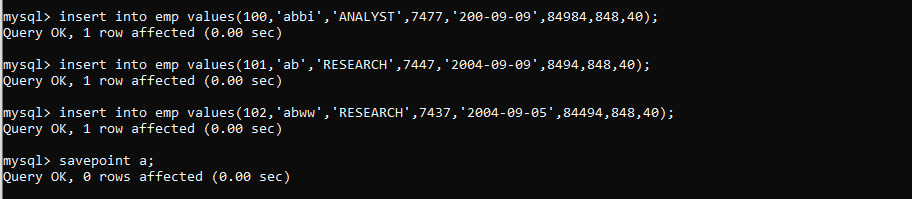


11. insert 3 records in dept and display all records from dept

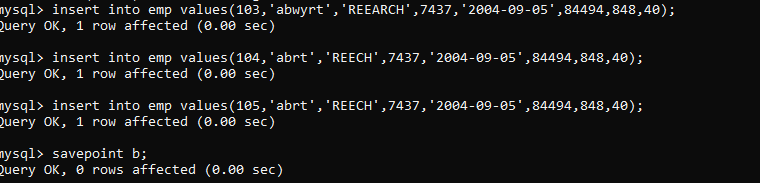


1. do the following

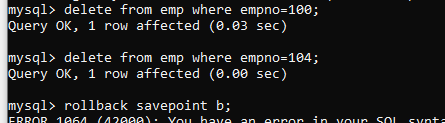
insert row in emp with empno 100 insert row in emp with empno 101 insert row in emp with empno 102 add savepoint A



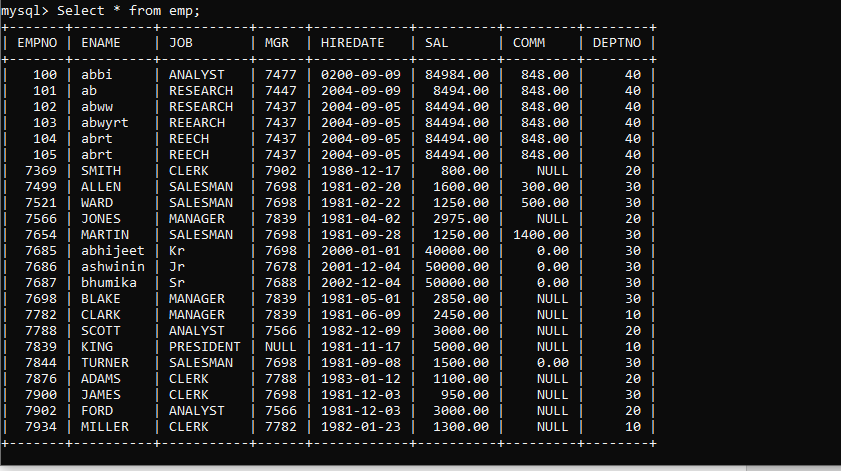
insert row in emp with empno 103 insert row in emp with empno 104 insert row in emp with empno 105 add savepoint B



delete emp with empno 100 delete emp with emp no 104 rollback upto svaepoint B



check what all records will appear in employee table



**PLSQL ASSIGNMENT 11**

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)

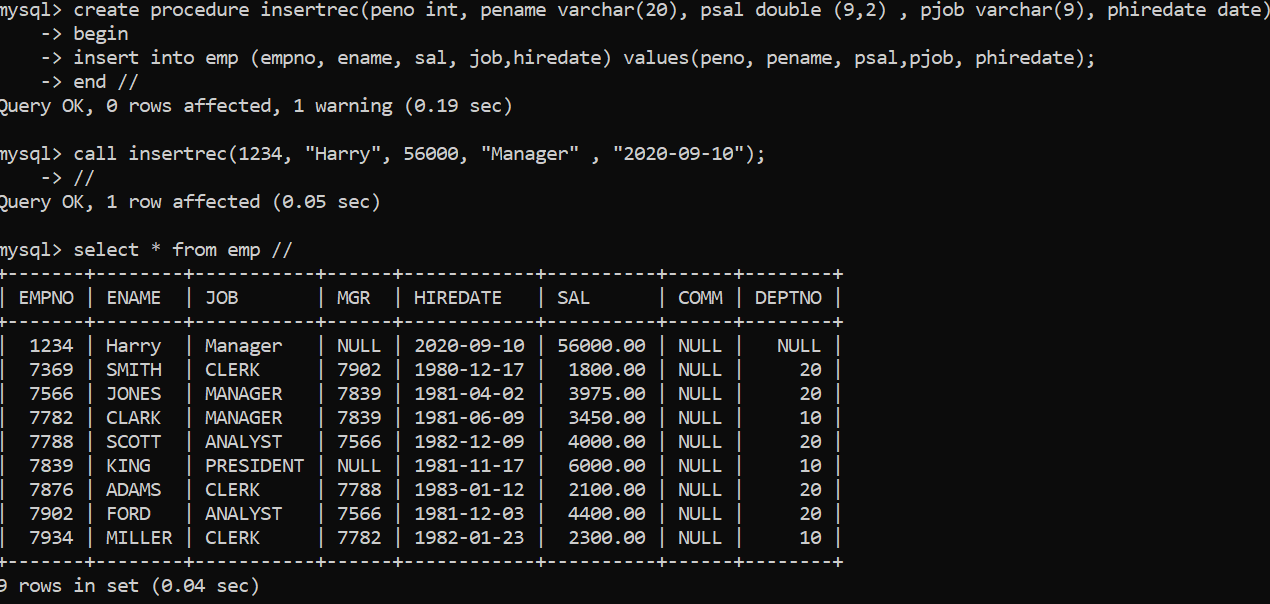
end//

**create procedure insertrec(peno int, pename varchar(20), psal double (9,2) , pjob varchar(9), phiredate date)**

**begin**

**insert into emp (empno, ename, sal, job,hiredate) values(peno, pename, psal,pjob, phiredate);**

**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

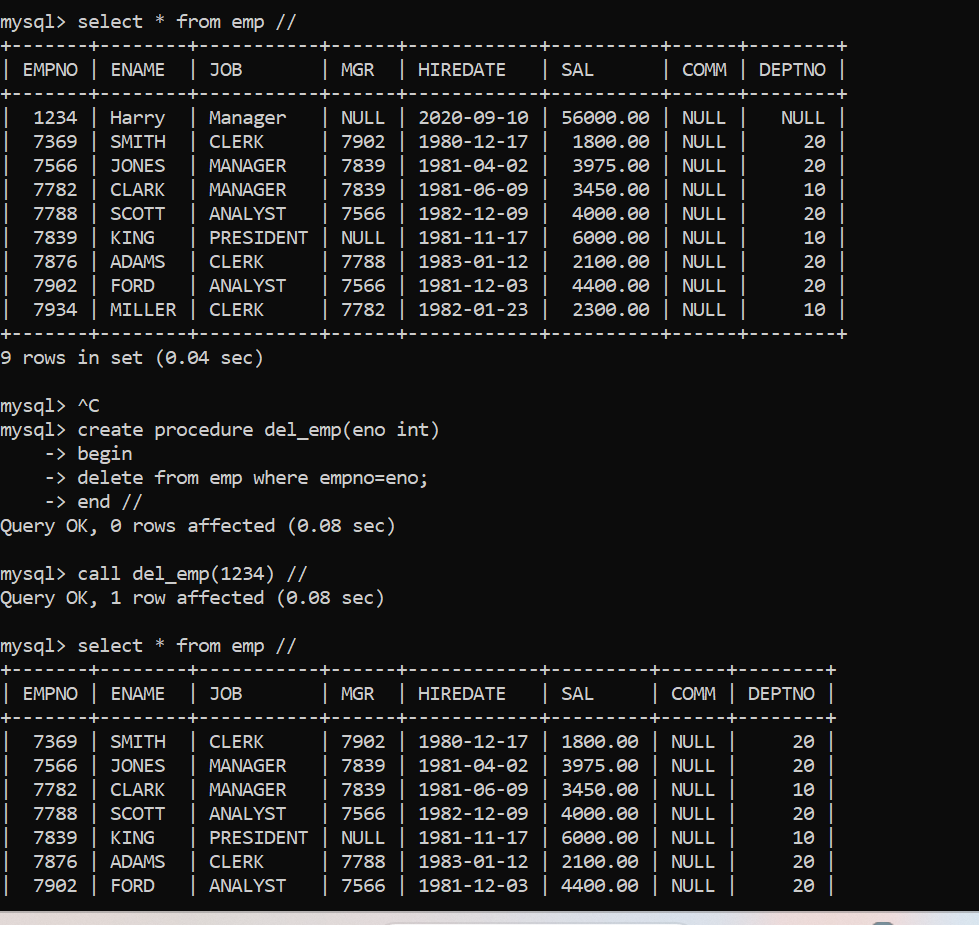
table

**create procedure del\_emp(eno int)**

**begin**

**delete from emp where empno=eno;**

**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 emp\_dept\_details(psal double)**

**begin**

**declare vename,vdname varchar(30);**

**declare vdeptno,vempno,vstop int default 0;**

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

**declare emp\_cur cursor for**

**select empno,ename,sal,deptno from emp where sal>psal;**

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

**open emp\_cur;**

**label1: loop**

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

**if vstop=1 then**

**leave label1;**

**end if;**

**select dname into vdname**

**from dept**

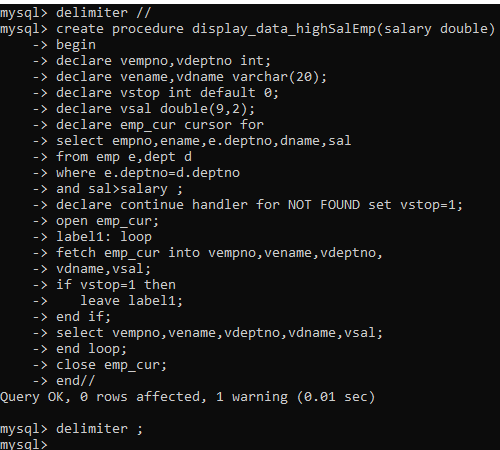
**where deptno=vdeptno;**

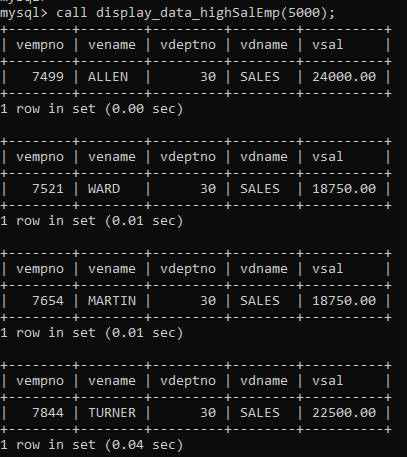
**select vempno,vename,vdeptno,vsal,vdname;**

**end loop;**

**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

**create procedure minmaxavg(in dno int, out minsal double(9,2), out maxsal double (9,2), out avgsal double(9,2))**

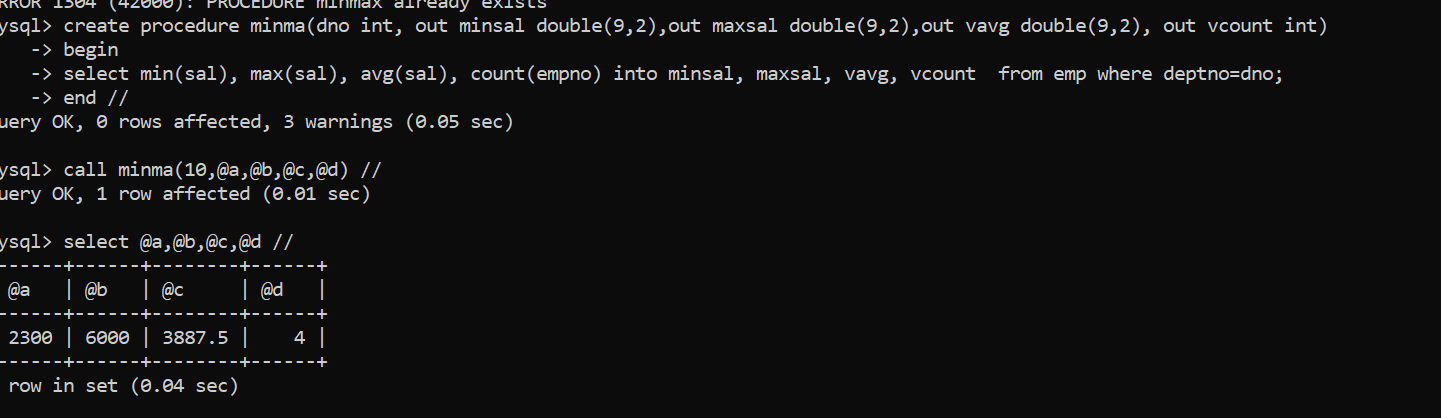
**begin**

**select min(sal), max(sal), avg(sal) into minsal , maxsal, avgsal from emp where deptno=dno;**

**end //**

**call minmax(20,@mini,@maxi,@avgg) //**

**select @mini,@maxi,@avgg //**



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

product,category and salesman table)

Sol:

delimiter //

create procedure display\_product\_details()

begin

declare vpid,vcid int ;

declare vpname,vcname,vsname varchar(20);

declare vstop int default 0;

declare product cursor for select prodid,pname from product;

declare continue handler for NOT FOUND set vstop=1;

open product;

label1:loop

fetch product into vpid,vpname;

if vstop=1 then

leave label1;

end if;

select p.prodid,p.pname,c.cid,c.cname,s.sname from

product p,category c, salesman s

where c.cid=p.catid and

s.sid = p.sid

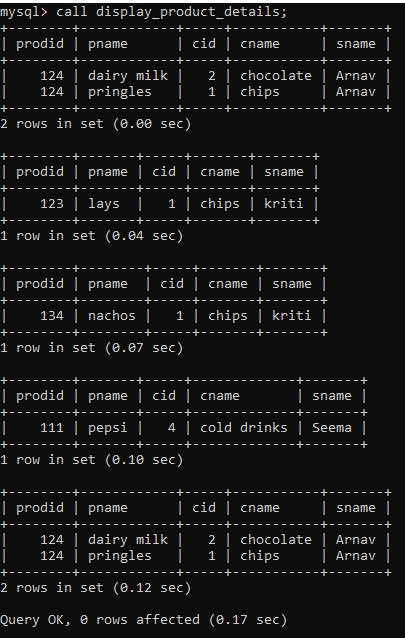
and p.prodid=vpid;

end loop;

close product;

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)

delimiter //

create procedure display\_vehicles(cnm varchar(20))

begin

declare pvid,pcid int;

declare pvname, pcname varchar(20);

declare vstop int default 0;

declare vehcust\_cur cursor for select custid,cname from customer\_v where cname=cnm;

declare continue handler for NOT FOUND set vstop=1;

open vehcust\_cur;

label1:loop

fetch vehcust\_cur into pcid,pcname;

if vstop=1 then

leave label1;

end if;

select c.custid,c.cname,v.vid,v.vname

from customer\_v c,vehicle v,cust\_vehicle cv

where cv.custid = c.custid

and cv.vid=v.vid

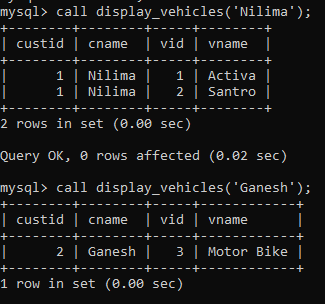
and cv.custid = pcid;

end loop;

close vehcust\_cur;

end//

delimiter ;



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.

**set global log\_bin\_trust\_function\_creators=1;**

**create function calcexp1(hdt date) returns int**

**begin**

**return floor(datediff(curdate(),hdt)/365);**

**end //**

**create function additional\_allowan(psal double(9,2) , hdt date) returns double**

**begin**

**declare vexp\_allow double(9,2) default 0;**

**declare vexp int;**

**set vexp=calcexp1(hdt);**

**if vexp>=36 and vexp<40 then set vexp\_allow=psal\*0.10;**

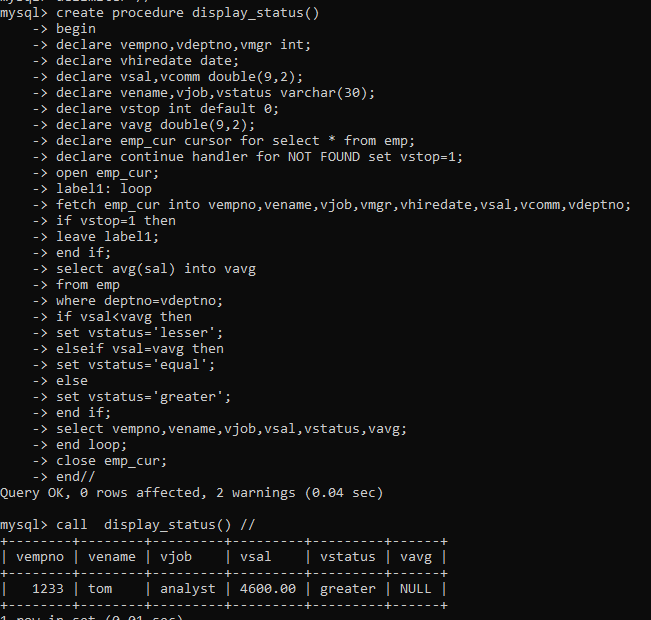
**elseif vexp>=40 and vexp<42 then set vexp\_allow=psal\*0.20;**

**elseif vexp>=42 then set vexp\_allow=psal\*0.30;**

**end if;**

**return psal+vexp\_allow+0.15\*psal+ 0.20\*psal+0.08\*psal;**

**end //**



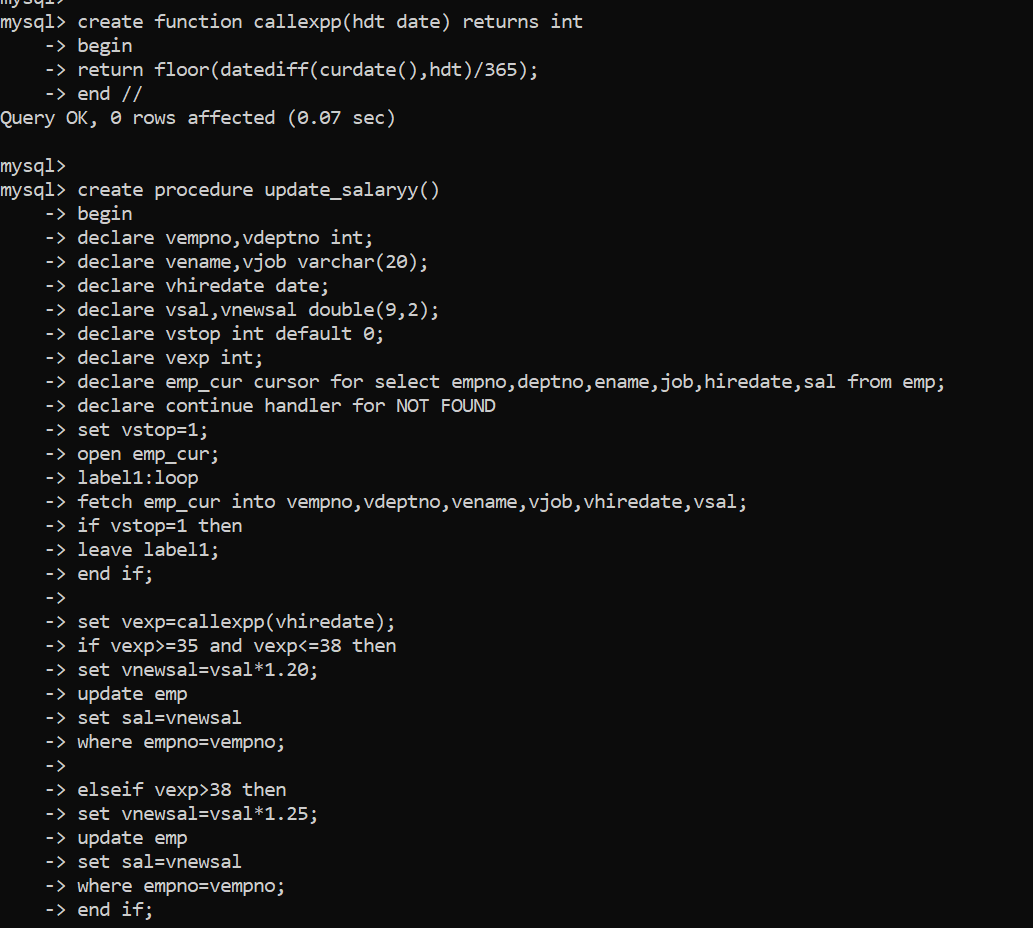
8. Write a procedure to update salary in emp table based on following rules.

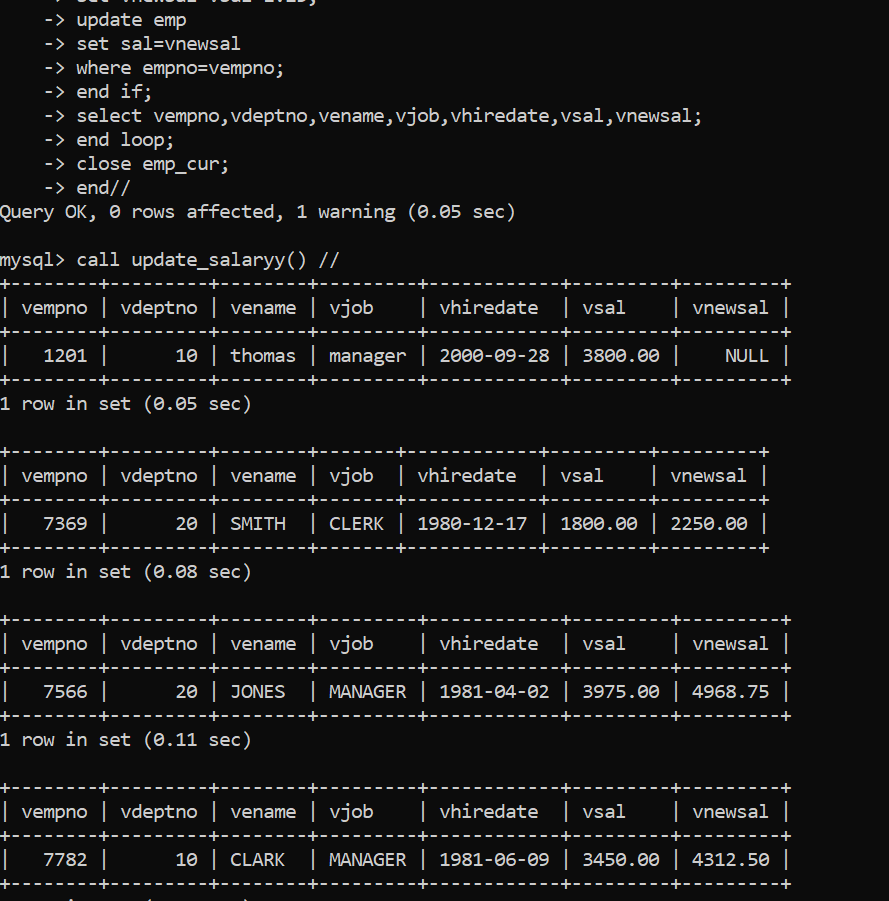
Exp< =35 then no Update

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

Exp> 38 then 25% of salary

**delimiter //  
create procedure update\_sal()  
begin  
declare vempno,vdeptno int;  
declare vename,vjob varchar(20);  
declare vhiredate date;  
declare vsal,vnewsal double(9,2);  
declare vstop int default 0;  
declare vexp int;  
declare emp\_cur cursor for select empno,deptno,ename,job,hiredate,sal from emp;  
declare continue handler for NOT FOUND  
set vstop=1;  
open emp\_cur;  
label1:loop  
fetch emp\_cur into vempno,vdeptno,vename,vjob,vhiredate,vsal;  
if vstop=1 then  
leave label1;  
end if;  
  
set vexp=cal\_experience(vhiredate);  
if vexp>=35 and vexp<=38 then  
set vnewsal=vsal\*1.20;  
update emp  
set sal=vnewsal  
where empno=vempno;  
  
elseif vexp>38 then  
set vnewsal=vsal\*1.25;  
update emp  
set sal=vnewsal  
where empno=vempno;  
end if;  
select vempno,vdeptno,vename,vjob,vhiredate,vsal,vnewsal;  
end loop;  
close emp\_cur;  
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 table emp\_allowance(

empno int,

ename varchar(20),

hiredate date,

experience int,

allowance decimal(9,2));

create table emp\_allowance( empno int, ename varchar(20), hiredate date, experience int, allowance double(9,2));

create procedure inser\_recordd(eno int, nm varchar(20), vhiredate date)

begin

declare vexper int;

declare vallow double(9,2);

set vexper=calcexp2(vhiredate);

set vallow =vexper\*3000;

insert into emp\_allowance (empno, ename, hiredate,experience, allowance) values (eno, nm, vhiredate, vexper, vallow);

end //

create function calcexp2(hdt date)

-> returns int

-> begin

-> return floor(datediff(curdate(),hdt)/365);

-> end//

create function additional\_allowan(psal double(9,2) , hdt date) returns double

begin

declare vexp\_allow double(9,2) default 0;

declare vexp int;

set vexp=calcexp1(hdt);

if vexp>=36 and vexp<40 then set vexp\_allow=psal\*0.10;

elseif vexp>=40 and vexp<42 then set vexp\_allow=psal\*0.20;

elseif vexp>=42 then set vexp\_allow=psal\*0.30;

end if;

return psal+vexp\_allow+0.15\*psal+ 0.20\*psal+0.08\*psal;

end //

Query OK, 0 rows affected (0.02 sec)

create function calcexp2(hdt date)

-> returns int

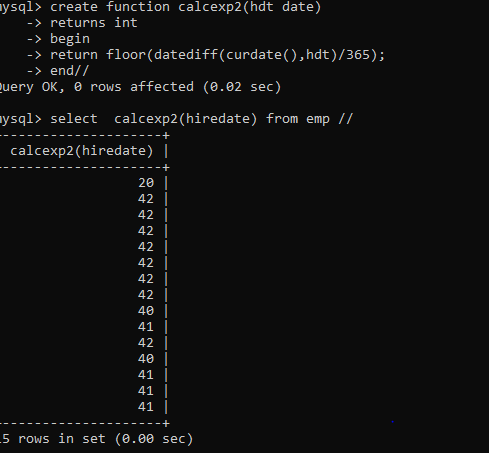
-> begin

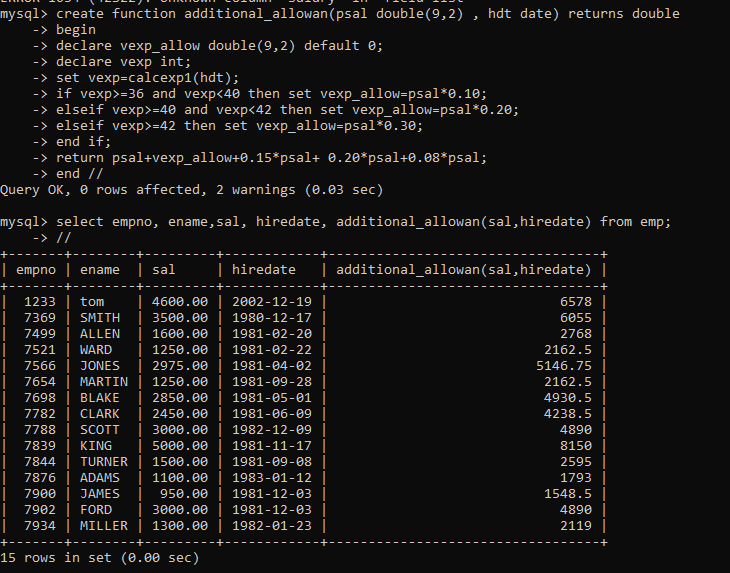
-> return floor(datediff(curdate(),hdt)/365);

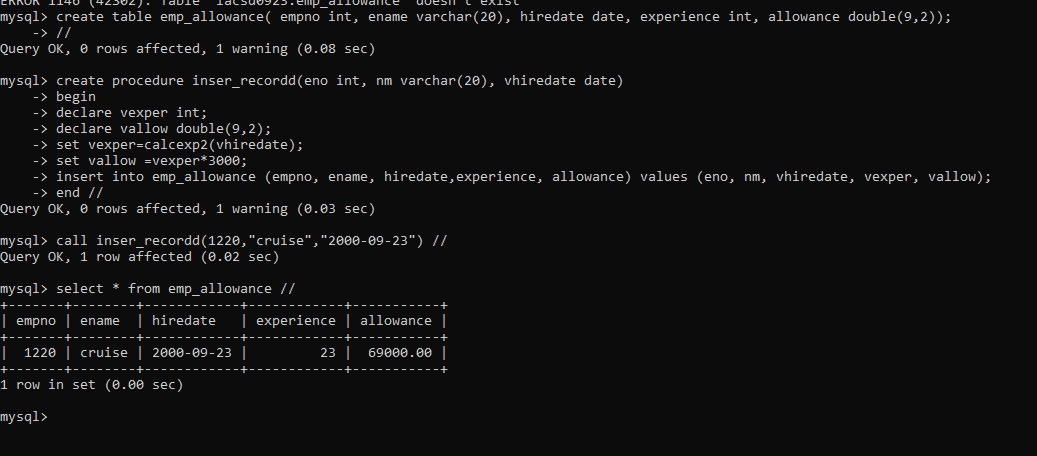
-> end//

Query OK, 0 rows affected (0.02 sec)

mysql> select calcexp2(hiredate) from emp //







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 cal\_sal(salary double,hdt date) returns double

begin

declare vexp int;

declare vsal double(9,2) default 0;

set vexp=calexp(hdt);

if vexp>35 and vexp<=38 then

set vsal = salary\*1.10;

elseif vexp>38 and vexp<=42 then

set vsal = salary\*1.20;

elseif vexp>42 then

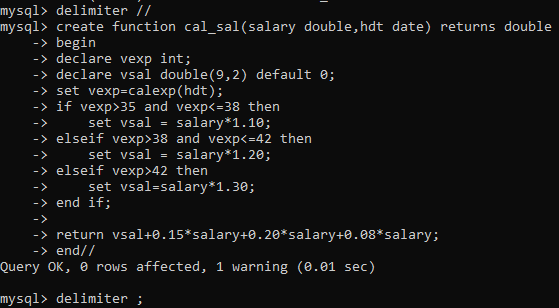
set vsal=salary\*1.30;

end if;

return vsal+0.15\*salary+0.20\*salary+0.08\*salary;

end//

delimiter ;



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

use function written in question 10)

delimiter //

create procedure cost\_to\_company()

begin

declare vempno int;

declare vename varchar(20);

declare vsalary,newsal double(9,2);

declare vhiredate date;

declare vstop int default 0;

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

declare continue handler for NOT FOUND set vstop=1;

open ctc\_cur;

label1:loop

fetch ctc\_cur into vempno,vename,vsalary,vhiredate;

if vstop=1 then

leave label1;

end if;

set newsal=cal\_sal(vsalary,vhiredate);

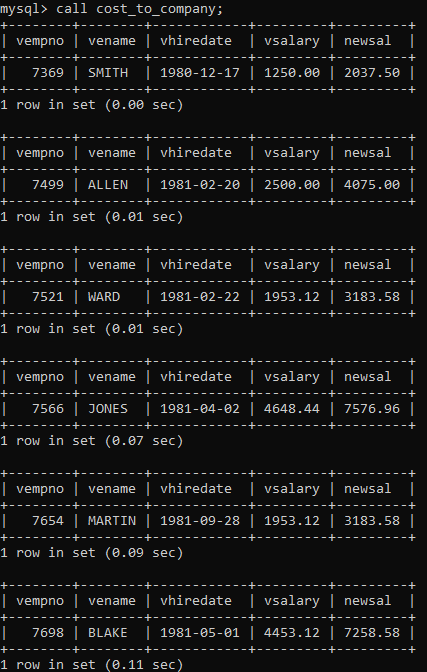
select vempno,vename,vhiredate,vsalary,newsal;

end loop;

close ctc\_cur;

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)

)

(note :

execute procedure written in Q8 and

check the entries in EMP\_back table after execution of the procedure)

delimiter //

create trigger monitor\_emp\_update after update on emp

for each row

begin

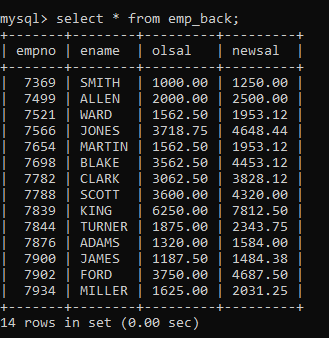
insert into emp\_back values(OLD.empno,OLD.ename,OLD.sal,NEW.sal);

end//

delimiter ;

After executing procedure from Q8:

call update\_sal;



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 em1paud(empno int , oldename varchar(20), newname varchar(20), action varchar(20) , uname varchar(20), chdate date) //

create trigger insem1 after insert on emp for each row

begin

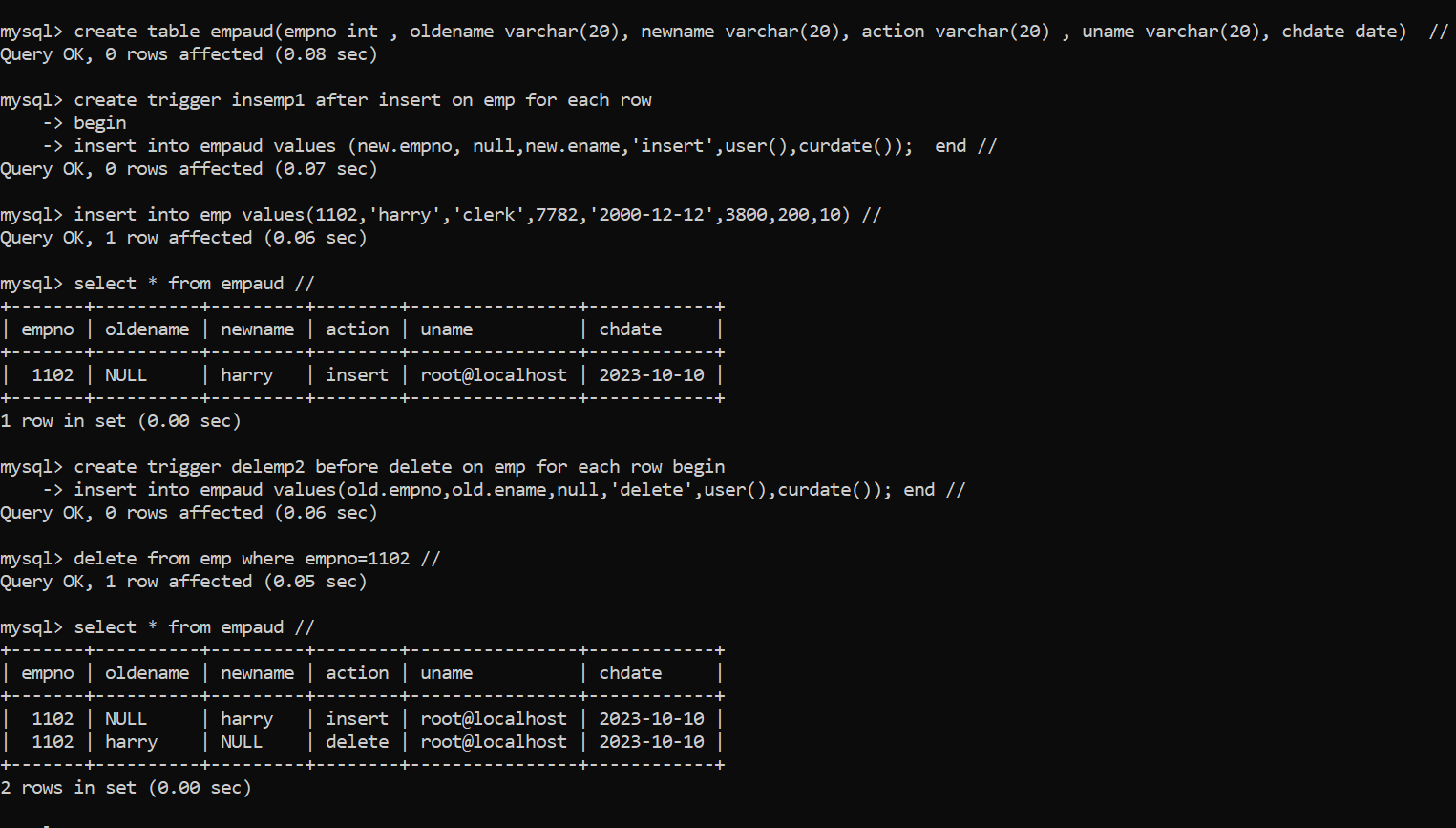
insert into em1paud values (new.empno, null,new.ename,'insert',user(),curdate()); end //

insert into emp values(11102,'harry','clerk',7782,'2000-12-12',3800,200,10) //

create trigger delemp2 before delete on emp for each row begin

insert into empaud values(old.empno,old.ename,null,'delete',user(),curdate()); end //

delete from emp where empno=1102 //



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,

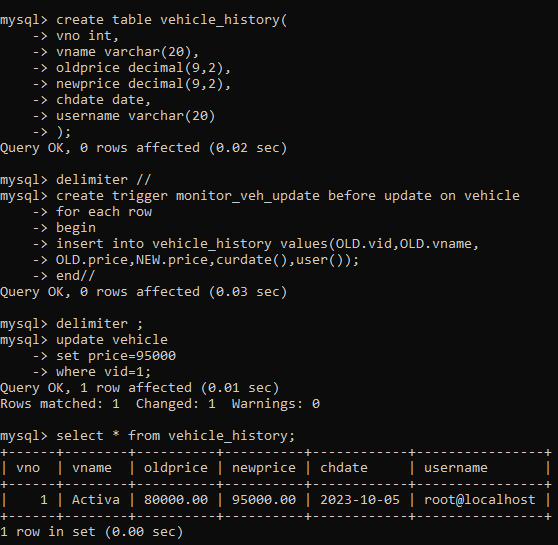
vname varchar(20),

oldprice decimal(9,2),

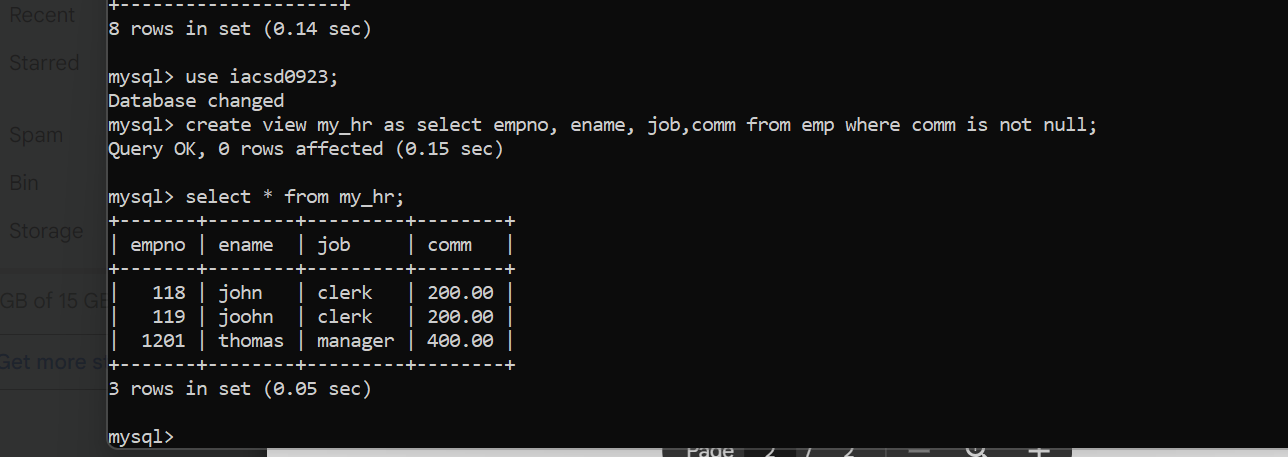
newprice decimal(9,2),

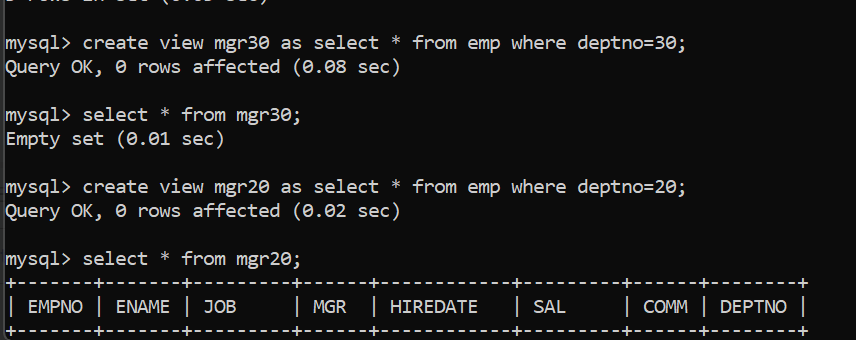
chdate date,

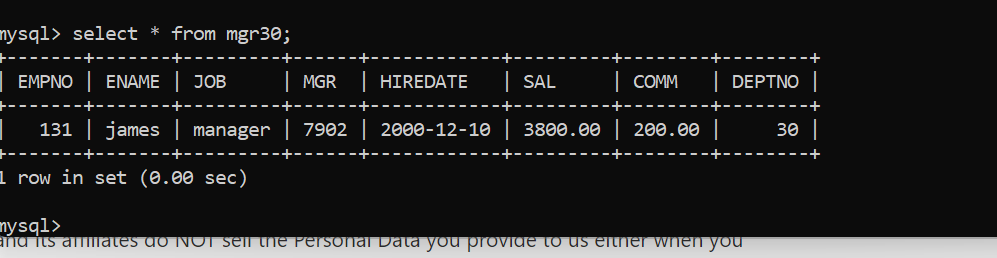
username varchar(20)



Views







MONGODB

1. Write a MongoDB query to display all the documents in the collection restaurants

2. Write a MongoDB query to display the fields restaurant\_id, name, borough and cuisine for all the documents in the collection restaurant.

3. Write a MongoDB query to display the fields restaurant\_id, name, borough and cuisine, but exclude the field \_id for all the documents in the collection restaurant.

4. Write a MongoDB query to display the fields restaurant\_id, name, borough and zip code, but exclude the field \_id for all the documents in the collection restaurant.

5. Write a MongoDB query to display all the restaurant which is in the borough Bronx

6. Write a MongoDB query to display the first 5 restaurant which is in the borough Bronx.

7.Write a MongoDB query to display the next 5 restaurants after skipping first 5 which are in the borough Bronx.

8. Write a MongoDB query to find the restaurants who achieved a score more than 90.

9. Write a MongoDB query to find the restaurants that achieved a score, more than 80 but less than 100.

10. Write a MongoDB query to find the restaurants which locate in latitude value less than -95.754168.

11. Write a MongoDB query to find the restaurants that do not prepare any cuisine of 'American' and their grade score more than 70 and latitude less than -65.754168.

12. Write a MongoDB query to find the restaurants which do not prepare any cuisine of 'American' and achieved a score more than 70 and located in the longitude less than -65.754168.

13. Write a MongoDB query to find the restaurants which do not prepare any cuisine of 'American ' and achieved a grade point 'A' not belongs to the borough Brooklyn. The document must be displayed according to the cuisine in descending order.

14. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Wil' as first three letters for its name.

15. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'ces' as last three letters for its name.

16. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Reg' as three letters somewhere in its name.

17. Write a MongoDB query to find the restaurants which belong to the borough Bronx and prepared either American or Chinese dish.

Answers:

1:db.restaurent.find()

2: db.restaurent.find({},{restaurent\_id:1,name:1,borough:1,cuisine:1})

3: db.restaurent.find({},{restaurent\_id:1,name:1,borough:1,cuisine:1,\_id:0})

4: db.restaurent.find({},{restaurent\_id:1,name:1,borough:1,'address.zipcode':1,\_id:0})

5: db.restaurent.find({borough:/^[Bb]ronx/})

6: db.restaurent.find({borough:/^[Bb]ronx/}).limit(5)

7: db.restaurent.find({borough:/^[Bb]ronx/}).limit(5).skip(5)

8: db.restaurent.find({'grades.score':{$gt:90}})

9: db.restaurent.find({'grades.score':{$gt:80,$lt:100}})

10: db.restaurent.find({'address.coord.0':{$lt:95.754168}})

11: db.restaurent.find({cuisine:{$ne:'American'},'grades.score':{$gt:70},'address.coord.0':{$lt:-65.754168}})

12:

db.restaurent.find({cuisine:{$ne:'American'},'grades.score':{$gt:70},'address.coord.0':{$lt:65.754168}

})

13:db.restaurent.find({$and:[{cuisine:{$ne:'American'}},{'grades.grade':{$eq:'A'}},{'borough':{$ne:'Brooklyn'}}]}).sort({'cuisine':-1})

14:

db.restaurent.find({name:/^Wil.\*/},{restaurent\_id:1,name:1,borough:1,cuisine:1,\_id:0})

15:

db.restaurent.find({name:/.\*ces$/},{restaurent\_id:1,name:1,borough:1,cuisine:1})

16:

db.restaurent.find({name:/.\*Reg.\*/},{restaurent\_id:1,name:1,borough:1,cuisine:1})

17:

db.restaurent.find({borough:'Bronx',cuisine:{$in:['American','Chinese']}})

18. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which belong to the borough Staten Island or Queens or Bronxor Brooklyn.

19. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which are not belonging to the borough Staten Island or Queens or Bronxor Brooklyn.

20. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which achieved a score which is not more than 10.

21. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which prepared dish except 'American' and 'Chinees' or restaurant's name begins with letter 'Wil'.

22. Write a MongoDB query to find the restaurant Id, name, and grades for those restaurants which achieved a grade of "A" and scored 11 on an ISODate "2014-08-11T00:00:00Z" among many of survey dates

23. Write a MongoDB query to find the restaurant Id, name and grades for those restaurants where the 2nd element of grades array contains a grade of "A" and score 9 on an ISODate "2014-08-11T00:00:00Z".

24. Write a MongoDB query to find the restaurant Id, name, address and geographical location for those restaurants where 2nd element of coord array contains a value which is more than 42 and upto 52

25. Write a MongoDB query to arrange the name of the restaurants in ascending order along with all the columns.

26. Write a MongoDB query to arrange the name of the restaurants in descending along with all the columns.

27. Write a MongoDB query to arranged the name of the cuisine in ascending order and for that same cuisine borough should be in descending order.

28. Write a MongoDB query to know whether all the addresses contains the street or not.

29. Write a MongoDB query which will select all documents in the restaurants collection where the coord field value is Double.

30. Write a MongoDB query which will select the restaurant Id, name and grades for those restaurants which returns 0 as a remainder after dividing the score by 7.

31. Write a MongoDB query to find the restaurant name, borough, longitude and attitude and cuisine for those restaurants which contains 'mon' as three letters somewhere in its name.

32. Write a MongoDB query to find the restaurant name, borough, longitude and latitude and cuisine for those restaurants which contain 'Mad' as first three letters of its name.

18:

db.restaurent.find({$or:[{borough:'Stalen Island'},{borough:'Bronxor Brooklyn'},{borough:'Queens'}]},{restaurent\_id:1,name:1,borough:1,cuisine:1})

19:

db.restaurent.find({borough:{$nin:['Stalen Island','Queens','Bronxor Brooklyn']}},{restaurent\_id:1,name:1,cuisine:1})

20:

db.restaurent.find({'grades.score':{$lte: 10}}, {restaurant\_id:1,name:1, borough:1,cuisine:1})

21:

db.restaurent.find({$nor: [{cuisine: {$in: ['American','Chinese']}},{name: /^Wil.\*/}]},{restaurant\_id:1, name:1,borough:1,cuisine:1})

22:

db.restaurent.find({grades: {$elemMatch: {date: ISODate('2014-08-11T00:00:00Z'), grade:'A', score:11}}},{restaurant\_id:1,name:1,grades:1})

23:

db.restrestaurent.find({$and: [{'grades.1.grade':'A'}, {'grades.1.score': 9}, {'grades.1.date': ISODate('2014-08-11T00:00:00Z')}]},{restaurant\_id:1, name:1, grades:1})

24:

db.restaurent.find({$and:[{'address.cord.1':{$gt:42}},{'address.cord.1':{$lt:52}}]},{restaurent\_id:1,name:1,address:1})

25:

db.restaurent.find({},{name:1}).sort({name:1})

26:

db.restaurent.find({},{name:1}).sort({name:-1})

27:

db.restaurent.find({},{cuisine:1,borough:1}).sort({cuisine:1,borough:-1})

28:

db.restaurent.find({'address.street':{/.\*Street.\*/}}).pretty()

db.restaurent.find({'address.street':{$ne:{/Street/}}}).pretty()

29:

db.restaurent.find({'address.coord':{$type:'double'}},{address:1})

30:

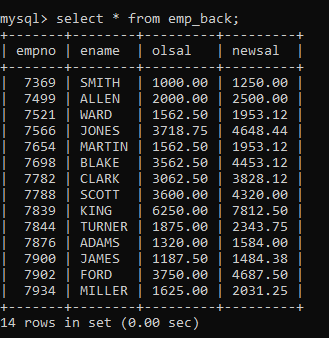
db.rest.find({grades: {$elemMatch: {score: {$mod:[7,0]}}}},{\_id:0, restaurant\_id:1, name:1, grades:1})

31:

db.rest.find({name: {$regex: /mon/}},{\_id:0, name:1, borough:1, "address.coord":1, cuisine:1})

32:

db.rest.find({name: {$regex: /^Mad.\*/}},{\_id:0, name:1, borough:1, "address.coord":1, cuisine:1})



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.

Sol:

create table empaudit(

empno int,

ename varchar(20),

username varchar(20),

chdate date,

action varchar(20)

);

delimiter //

create trigger emp\_data\_insert after insert on emp

for each row

begin

insert into empaudit values(NEW.empno,NEW.ename,user(),curdate(),'insert');

end//

delimiter ;

delimiter //

create trigger emp\_data\_delete after delete on emp

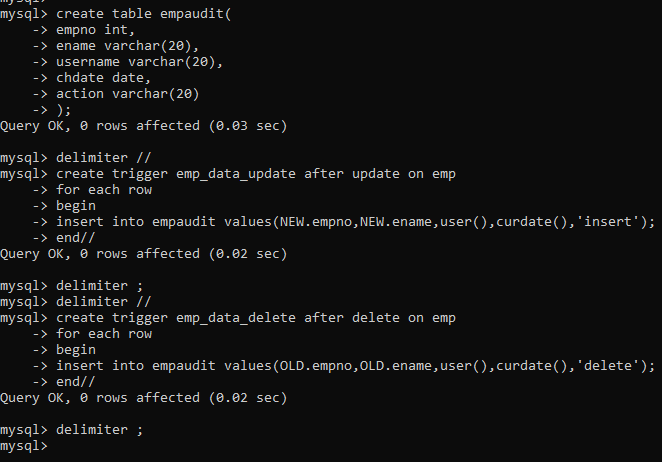
for each row

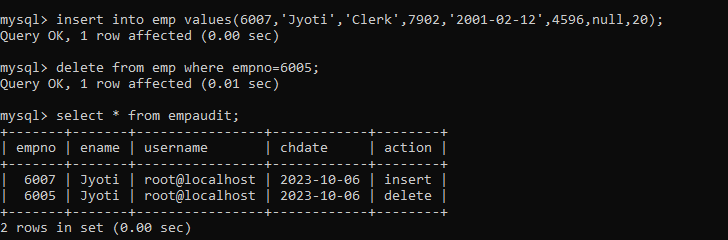
begin

insert into empaudit values(OLD.empno,OLD.ename,user(),curdate(),'delete');

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)

);

