**NAME :**  Kapil khanke

**ROLL NO :**  233160

**BATCH :** B

**DATABASE ASSIGNMENT - 1**

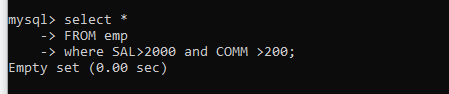
**Note : Use Emp, dept and salgrade table**

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

select \*

-> FROM emp

-> where SAL>2000 and COMM >200;

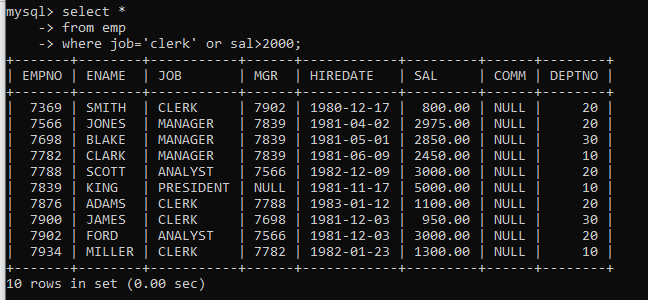


2. To list all record with job=’Clerk’ or sal>2000

mysql> select \*

-> from emp

-> where job='clerk' or sal>2000;

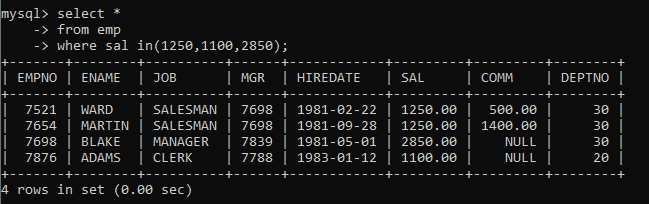


3. To list all the record with sal=1250 or 1100 or 2850

mysql> select \*

-> from emp

-> where sal in(1250,1100,2850);



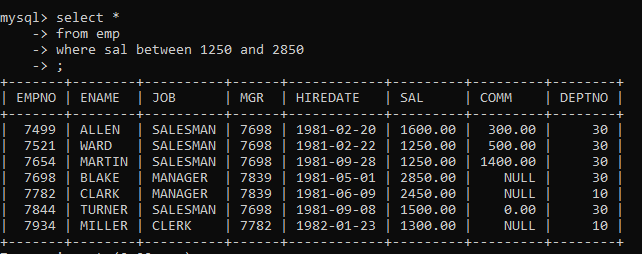
4. To list all employees with sal>1250 and <2850

mysql> select \*

-> from emp

-> where sal between 1250 and 2850

-> ;



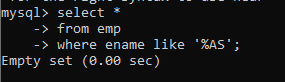
5. To list all employees with name ends with AS

mysql> select \*

-> from emp

-> where ename like '%AS';

Empty set (0.00 sec)

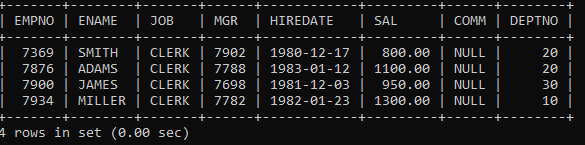


6. To list all employees with job starts with C and ends with K

mysql> select \*

-> from emp

-> where job like 'c%k';



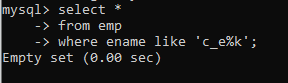
7. To list all employees with job contains L at third position and

M at third last position

mysql> select \*

-> from emp

-> where job like '\_\_l%m\_\_';

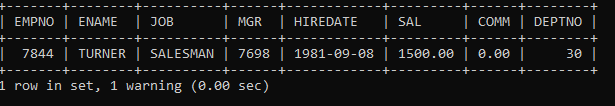


8. To list all the record with sal not equal to 1250 or 1100 or 2850

mysql> select \*

-> from emp

-> where sal not in (1250,1100,2850);

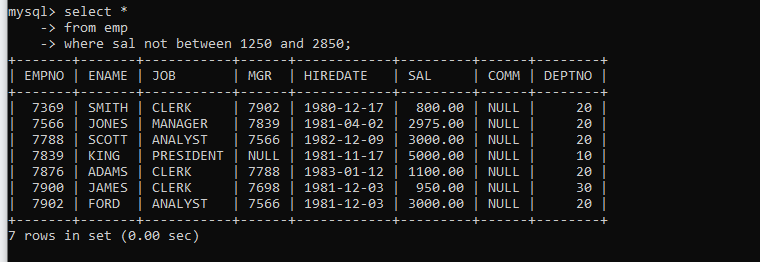


9. To list all employees with salnot >1250 and <2850

mysql> select \*

-> from emp

-> where sal not between 1250 and 2850;



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

mysql> select \*

-> from emp

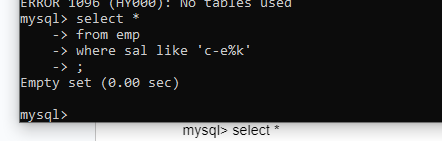
-> where ename like 'c\_e%k';

Or

select \*

from emp

where job regexp ‘^c.e.\*k$’

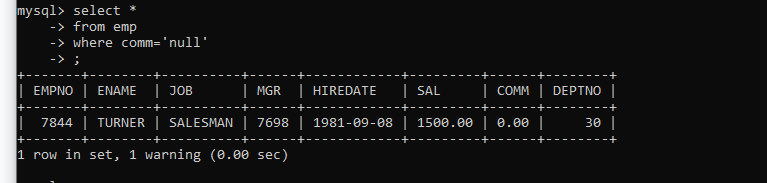


11. To list all rows with comm is null

mysql> select \*

-> from emp

-> where comm = 'null';



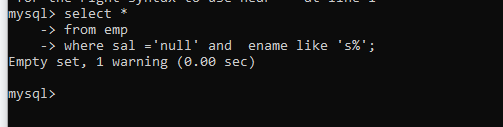
12. To list all employees with sal is null and name starts with ‘S’

mysql> select \*

-> from emp

-> where sal ='null' and ename like 's%';

Empty set, 1 warning (0.00 sec)



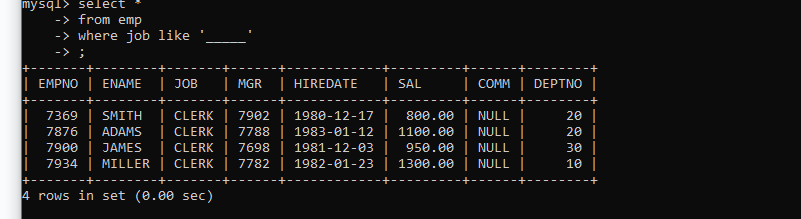
13. To list all employees with job contains 5 characters

select \*

-> from emp

-> where job like '\_\_\_\_\_'

-> ;



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

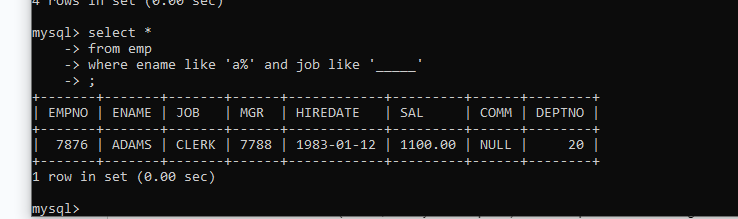
Contains 5 characters

select \*

-> from emp

-> where ename like 'a%' and job like '\_\_\_\_\_'

-> ;



**Q2. Solve the following**

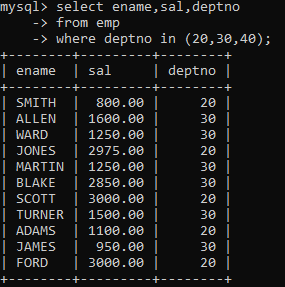
1. Retrieve the details (Name, Salary and dept no) of the emp who are working in

department code 20, 30 and 40.

mysql> select ename,sal,deptno

-> from emp

-> where deptno in (20,30,40);



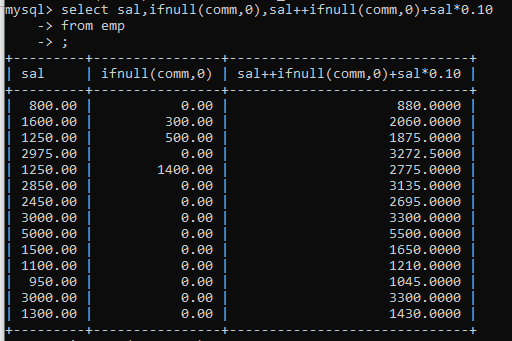
2. Display the total salary of all employees . Total salary will be calculated as

sal+comm+sal\*0.10

mysql> select sal,ifnull(comm,0),sal+ifnull(comm,0)+sal\*0.10

-> from emp

-> ;



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

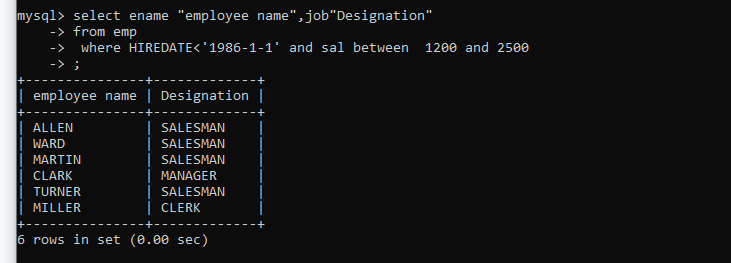
Headers.

mysql> select ename "employee name",job"Designation"

-> from emp

-> where HIREDATE<'1986-1-1' and sal between 1200 and 2500

-> ;



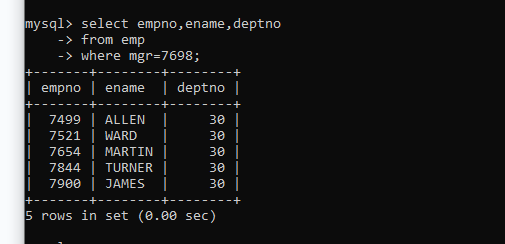
4. List the empno, name, and department number of the emp works under manager

with id 7698

> select empno,ename,deptno

-> from emp

-> where mgr=7698;



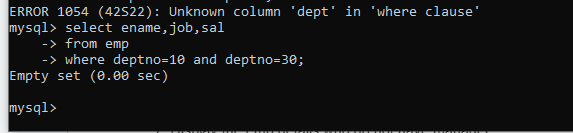
5. List the name, job, and salary of the emp who are working in departments 10 and

30.

mysql> select ename,job,sal

-> from emp

-> where dept=10 and dept=30;

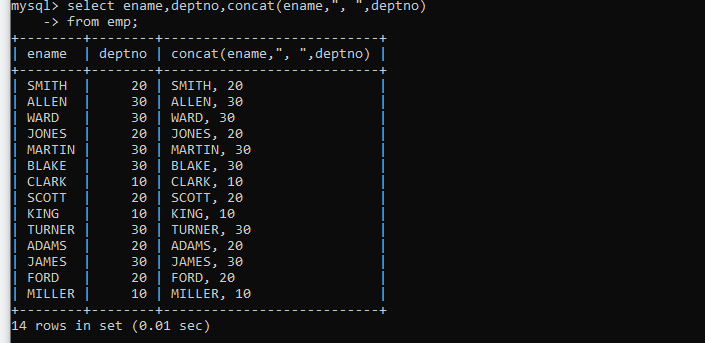


6. Display name concatenated with dept code separated by comma and space. Name

the column as ‘Emp info’.

select ename,deptno,concat(ename,", ",deptno)

-> from emp;

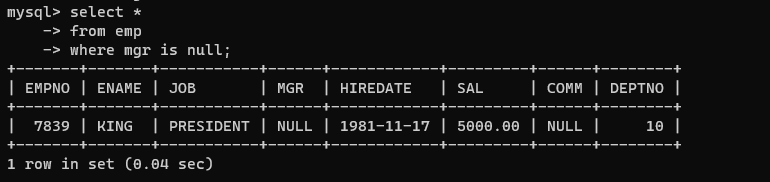


7. Display the emp details who do not have manager.

select \*

-> from emp

-> where mgr is null;



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

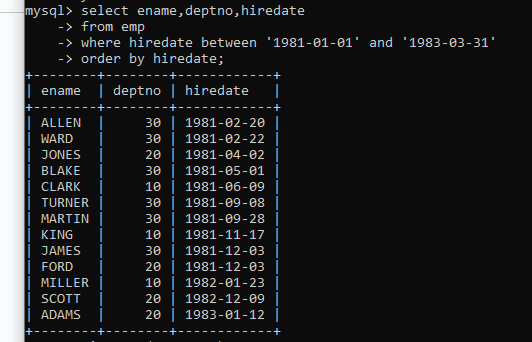
employee who were joined January 1, 1981 and March 31, 1983. Sort it based on date of joining (ascending).

select ename,deptno,hiredate

-> from emp

-> where hiredate between '1981-01-01' and '1983-03-31'

-> order by hiredate;



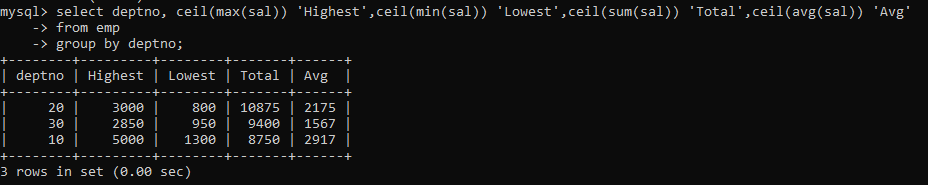
9. Display the employee details where the job contains word ‘AGE’ anywhere in the Job

select \*

-> from emp

-> where job regexp '.\*age.\*'

-> ;



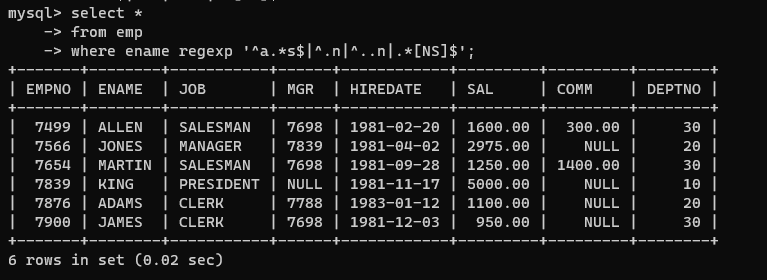
11. List the details of the employee , whose names start with ‘A’ and end with ‘S’ or

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

select \*

from emp

where ename regexp ‘^a.\*s$|^.n|^..n|.\*[NS]$’;



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

select \*

-> from emp

-> where ename regexp '.\*\_.\*';

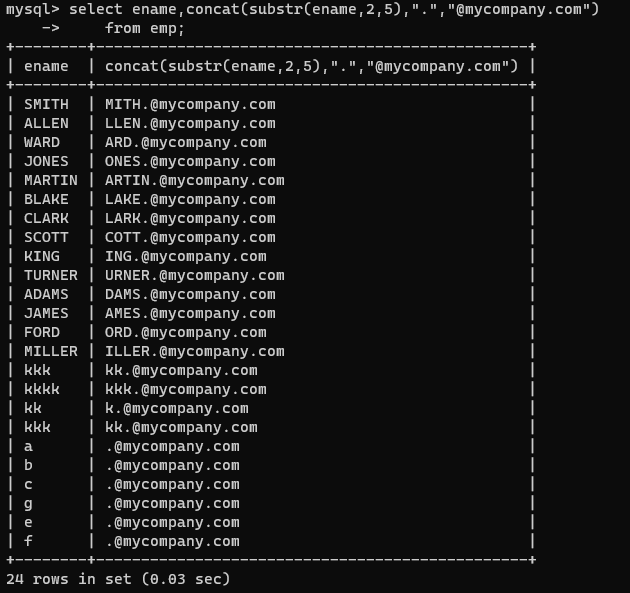
****

**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’

⇒select ename,concat(substr(ename,2,5),".","@mycompany.com")

from emp;

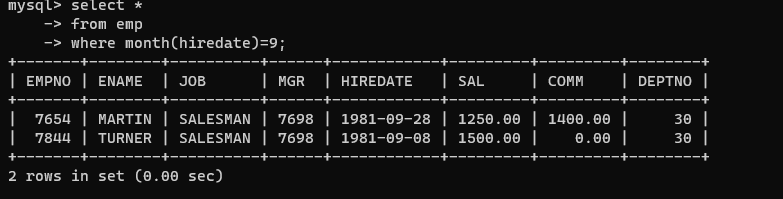


2. List all employees who joined in September.

select \*

-> from emp

-> where month(hiredate)=09;

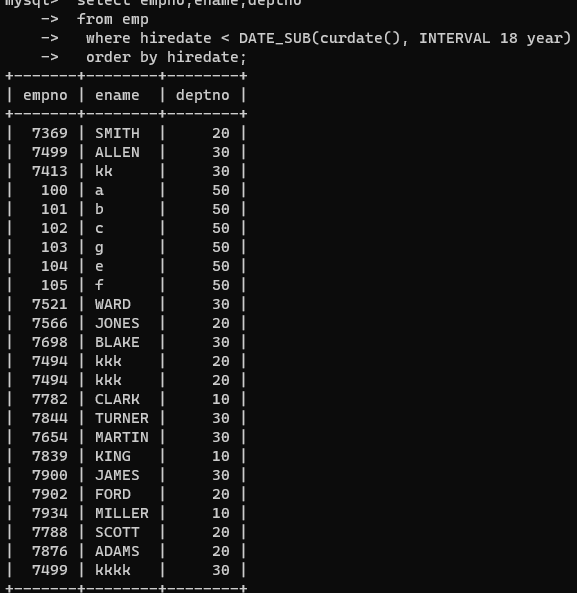


3. List the empno, name, and department number of the emp who have experience of 18 or more years and sort them based on their experience.

select empno,ename,deptno

-> from emp

-> where hiredate < DATE\_SUB(curdate(), INTERVAL 18 year)

-> order by hiredate;

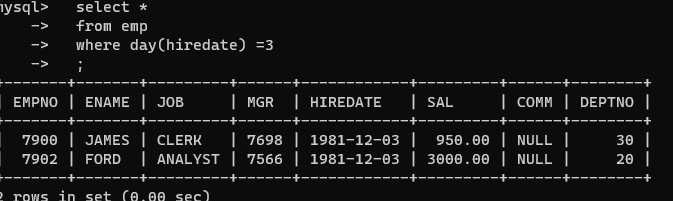
4. Display the employee details who joined on 3rd of any month or any year

select \*

-> from emp

-> where day(hiredate) =3

-> ;



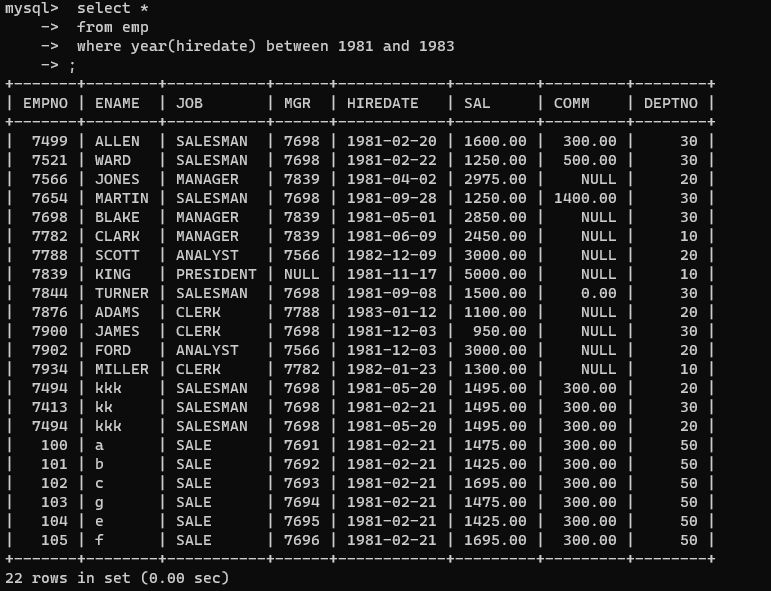
5. display all employees who joined between years 1981 to 1983.

select \*

-> from emp

-> where year(hiredate) between 1981 and 1983

-> ;



**Group functions**

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

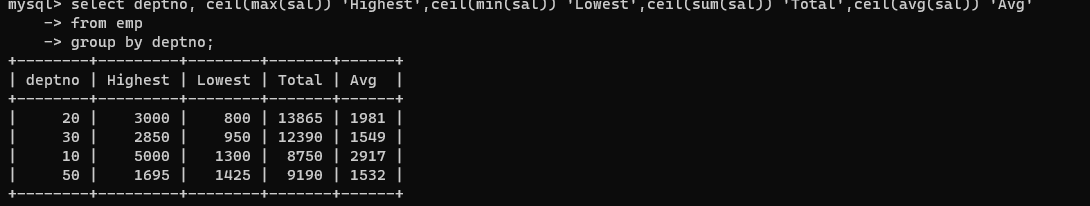
Maximum, Minimum, Total and Average respectively for each Department. Also round the

result to the nearest whole number.

mysql> select deptno, ceil(max(sal)) 'Highest',ceil(min(sal)) 'Lowest',ceil(sum(sal)) 'Total',ceil(avg(sal)) 'Avg'

-> from emp

-> group by deptno;



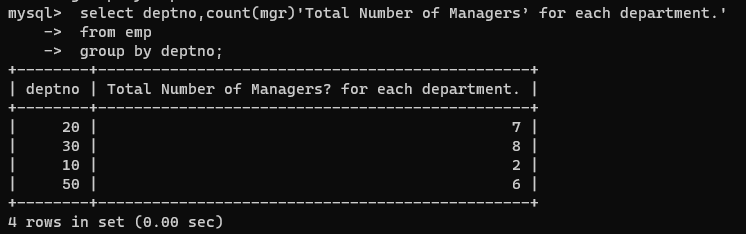
7. Display Department no and number of managers working in that department. Label the

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

select deptno,count(mgr)'Total Number of Managers’ for each department.'

-> from emp

-> group by deptno;



8. Get the Department number, and sum of Salary of all non managers where the sum is greater than 20000.

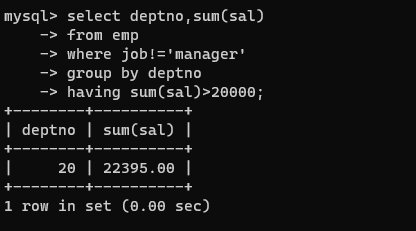
select deptno,sum(sal)

-> from emp

-> where job!='manager'

-> group by deptno

-> having sum(sal)>20000;



**ASSIGNMENT - 2**

Write SQL statement for the following

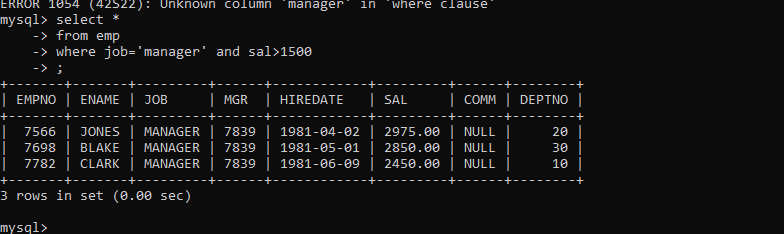
1. To find all managers with salary >1500

select \*

-> from emp

-> where job='manager' and sal>1500

-> ;



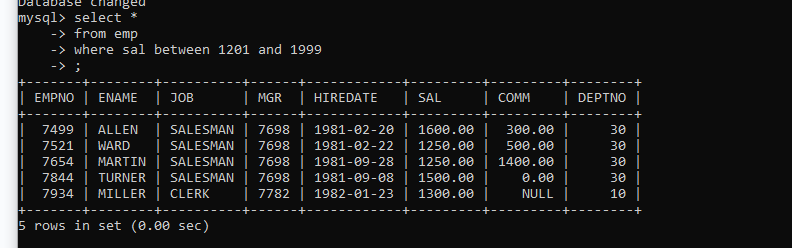
2. list all employees with sal >1200 and < 2000

select \*

-> from emp

-> where sal between 1201 and 1999

-> ;



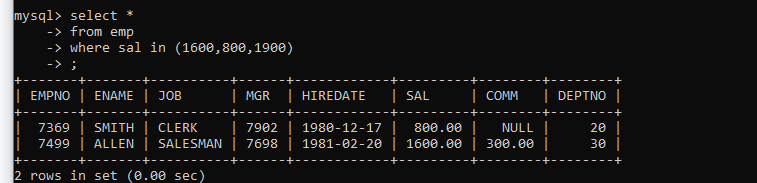
2. list all employees with sal is 1600 or sal is 800 or sal is 1900

select \*

-> from emp

-> where sal in (1600,800,1900)

-> ;



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

select \*

-> from emp

-> where ename regexp '^.\*R.$'

-> ;



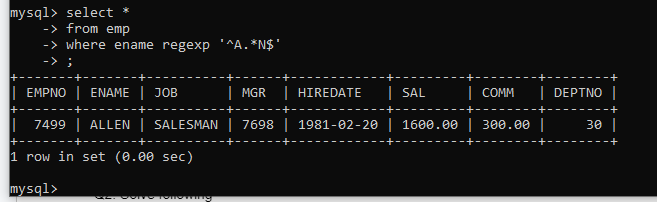
5. List all employees with name starts with A and ends with N

select \*

-> from emp

-> where ename regexp '^A.\*N$'

-> ;



Q2. Solve following

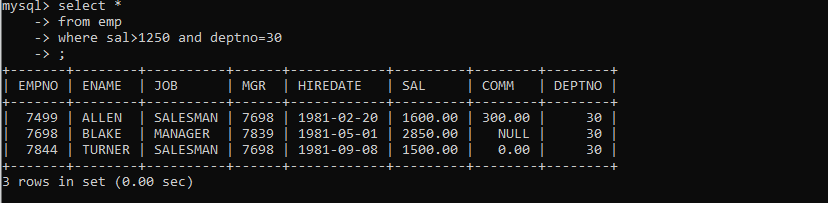
1. list all employees with salary > 1250 and dept no=30

select \*

-> from emp

-> where sal>1250 and deptno=30

-> ;

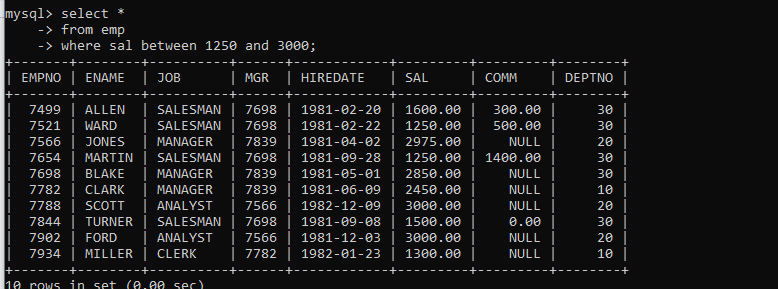


2. list all employees with salary >=1250 and <= 3000

select \*

-> from emp

-> where sal between 1250 and 3000;



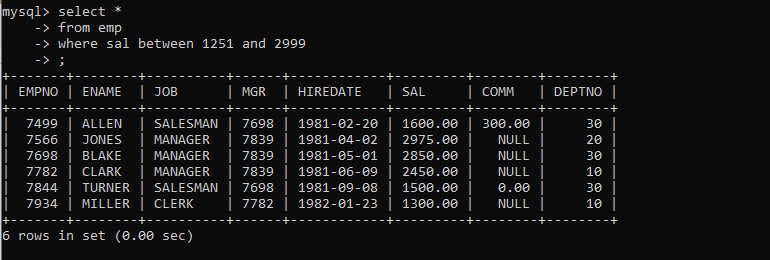
3. list all employees with salary >1250 and < 3000

select \*

-> from emp

-> where sal between 1251 and 2999

-> ;



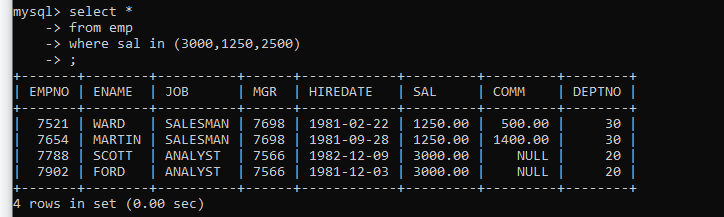
4. list all employees with salary either equal to 3000 or 1250 or 2500

select \*

-> from emp

-> where sal in (3000,1250,2500)

-> ;



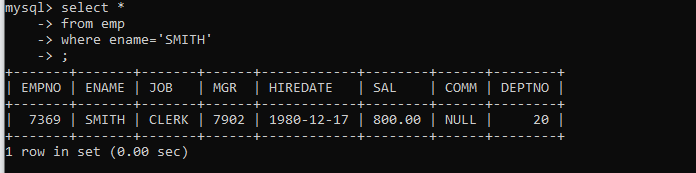
5. list all employee with name=SMITH

select \*

-> from emp

-> where ename='SMITH'

-> ;



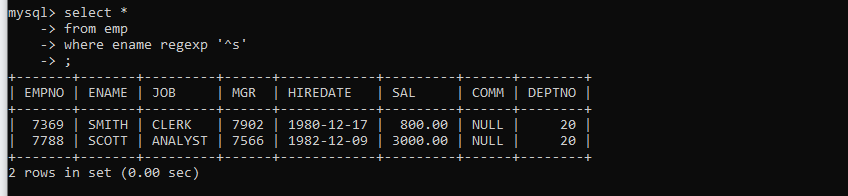
6. list all employees with name starting with S

mysql> select \*

-> from emp

-> where ename regexp '^s'

-> ;



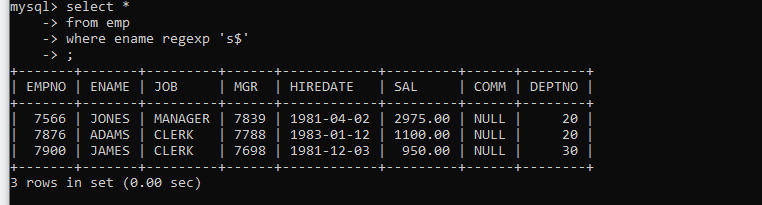
7. list all employees with name ending with S

> select \*

-> from emp

-> where ename regexp 's$'

-> ;



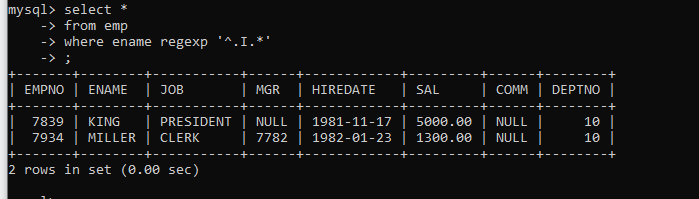
8. list all employees with name contains I at 2nd position

> select \*

-> from emp

-> where ename regexp '^.I.\*'

-> ;



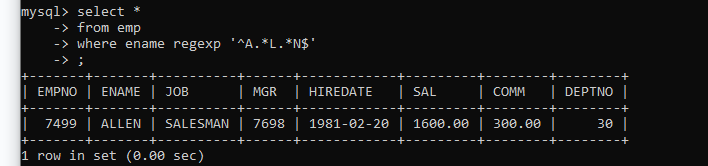
9. list all employees with name starts with A ends with N and somewhere in between L is there

> select \*

-> from emp

-> where ename regexp '^A.\*L.\*N$'

-> ;



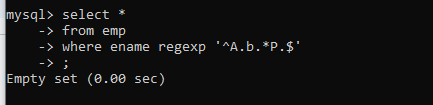
10. list all employees with name starts with A and B at 3 rd position and P at second last position

> select \*

-> from emp

-> where ename regexp '^A.b.\*P.$'

-> ;



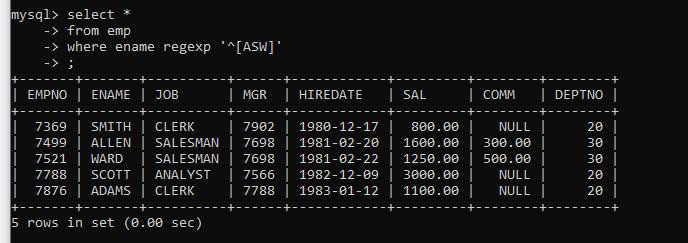
11. List all employees with name starts with either A or starts with S or starts with W

> select \*

-> from emp

-> where ename regexp '^[ASW]'

-> ;



practice Aggregate functions

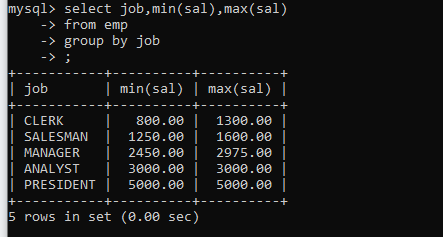
\12. find max sal and min sal for each job

> select job,min(sal),max(sal)

-> from emp

-> group by job

-> ;



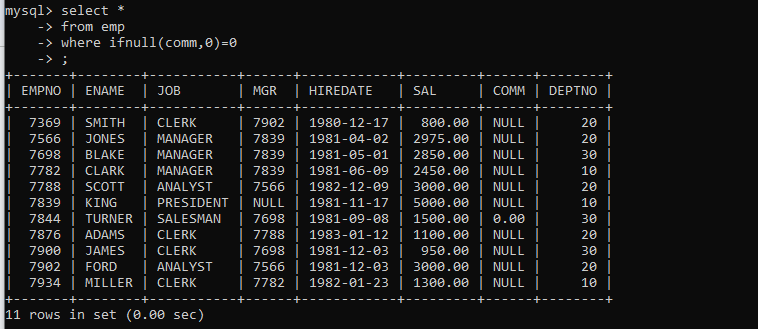
13. find how many employess have not received commission

> select \*

-> from emp

-> where ifnull(comm,0)=0

-> ;



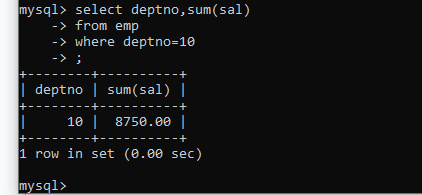
14. find sum of sal of all employees working in dept no 10

> select deptno,sum(sal)

-> from emp

-> where deptno=10

-> ;



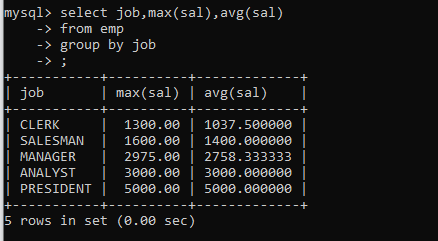
15. find maximum salary,average sal for each job in every department

> select job,max(sal),avg(sal)

-> from emp

-> group by job

-> ;



16. find max salary for every department if deptno is > 15 and arrange data in deptno order.

> select deptno,max(sal)

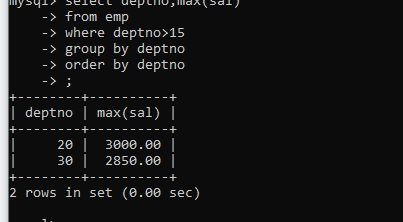
-> from emp

-> where deptno>15

-> group by deptno

-> order by deptno

-> ;



17. find sum salary for every department if sum is > 3000

select deptno,sum(sal)

-> from emp

-> group by deptno

-> having sum(sal)>3000

-> ;



18. list all department which has minimum 5 employees

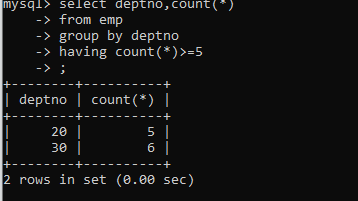
select deptno,count(\*)

-> from emp

-> group by deptno

-> having count(\*)>=5

-> ;



19. count how many employees earn salary more than 2000 in each job

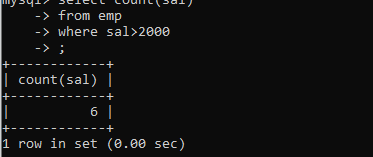
> select count(sal)

-> from emp

-> where sal>2000

-> group by job

-> ;

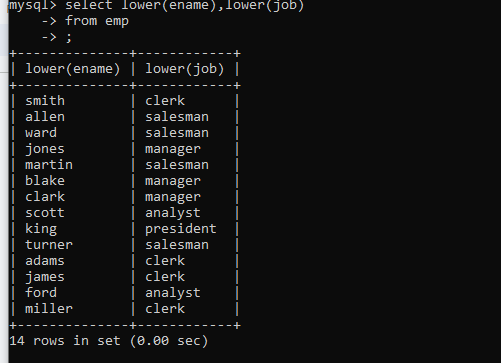


20. list all enames and jobs in small case letter

->select lower(ename),lower(job)

-> from emp

-> ;

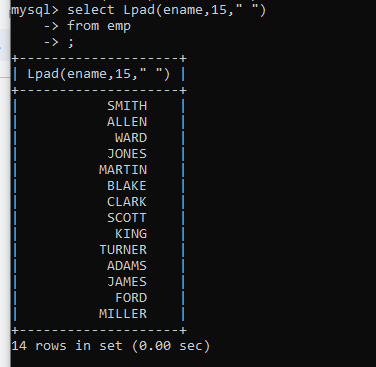


21. list all names and jobs so that the length of name should be 15 if it is smaller then add spaces to left

> select Lpad(ename,15," ")

-> from emp

-> ;



22. display min sal,max sal, average sal for all employees working under same manager

select mgr,min(sal),max(sal),avg(sal)

-> from emp

-> group by mgr

-> ;



23. find sum of total earnings(sal+comm), average of sal+comm for all employees who earn sal >

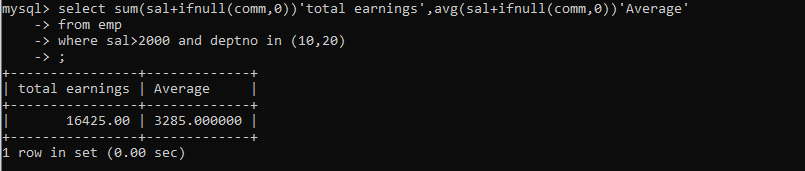
2000 and work in either dept no 10 or 20

> select sum(sal+ifnull(comm,0))'total earnings',avg(sal+ifnull(comm,0))'Average'

-> from emp

-> where sal>2000 and deptno in (10,20)

-> ;



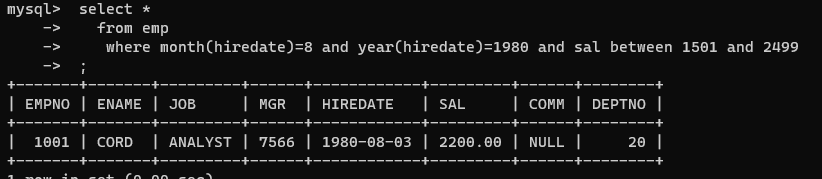
24. list all employees who joined in Aug 1980 and salary is >1500 and < 2500

select \*

-> from emp

-> where month(hiredate)=8 and year(hiredate)=1980 and sal between 1501 and 2499

-> ;



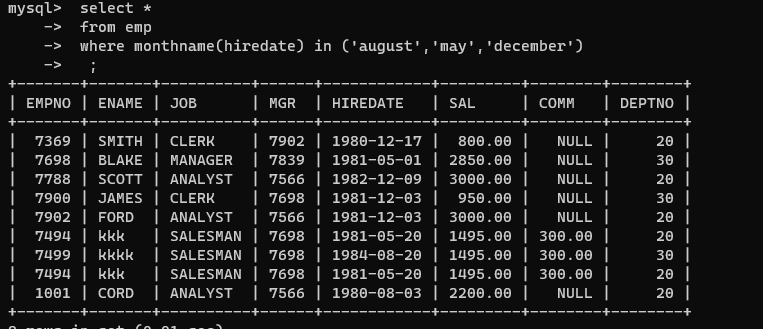
25. list all employees joined in either aug or may or dec

select \*

-> from emp

-> where monthname(hiredate) in ('august','may','december')

-> ;



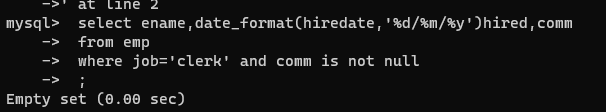
26. display name and hiredate in dd/mm/yy format for all employees whose job is clerk and they earn some commission

select ename,date\_format(hiredate,'%d/%m/%y')hired,comm

-> from emp

-> where job='clerk' and comm is not null

-> ;

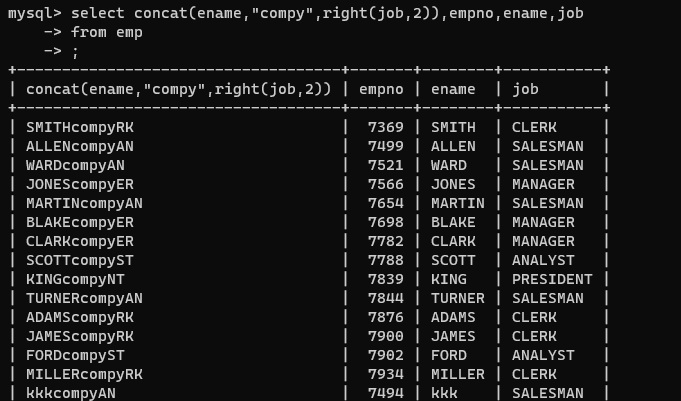


27. list empcode,empno,name and job for each employee. (note :empcode is 3 to 5 characters from name and last 2 characters of job)

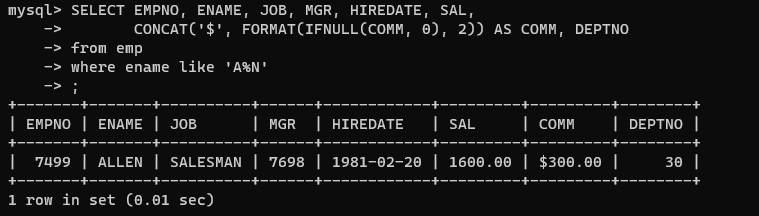
select concat(ename,"compy",right(job,2)),empno,ename,job

-> from emp

-> ;



28. display thousand separator and $ symbol for commission if it is null then display it as 0 for all employees whose name starts with A and ends with N



29. Display empid,name,sal,comm,remark Remark should base on following conditions

comm >= 600 "excellent Keep it up"

if it < 600 or not null "good"

otherwise "Need improvement"

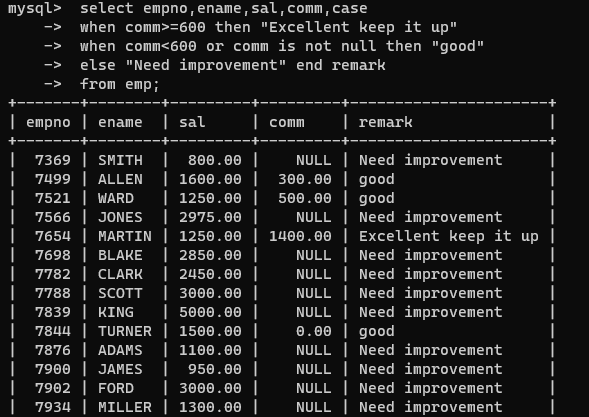
> select empno,ename,sal,comm,case

-> when comm>=600 then "Excellent keep it up"

-> when comm<600 or comm is not null then "good"

-> else "Need improvement" end remark

-> from emp;



30. Display empid, name, deptno and department name by using following conditions.

dept 10 then "Hr"

if 20 then "Admin"

if 30 then "accounts"

otherwise purchase

select empno,ename,deptno,case

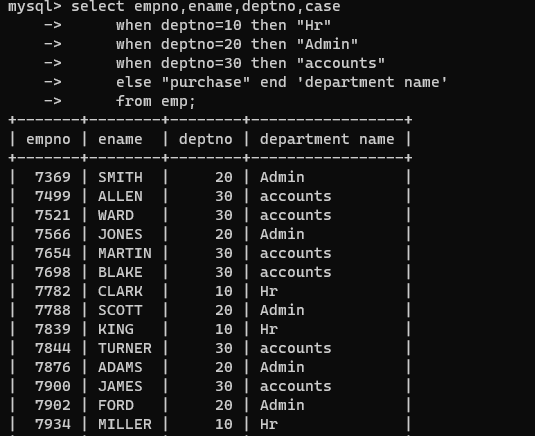
-> when deptno=10 then "Hr"

-> when deptno=20 then "Admin"

-> when deptno=30 then "accounts"

-> else "purchase" end 'department name'

-> from emp;



**Topic ----------------- create Table, DML , subquery and joins**

31. Practice creating following tables

create table mydept\_DBDA

(

deptid int(5) primary key,

dname varchar(20) not null unique,

dloc varchar(20)

)

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

create table myemployee

(

empno int(5) primary key,

fname varchar(20) not null,

mname varchar(20),

iname varchar(15) not null,

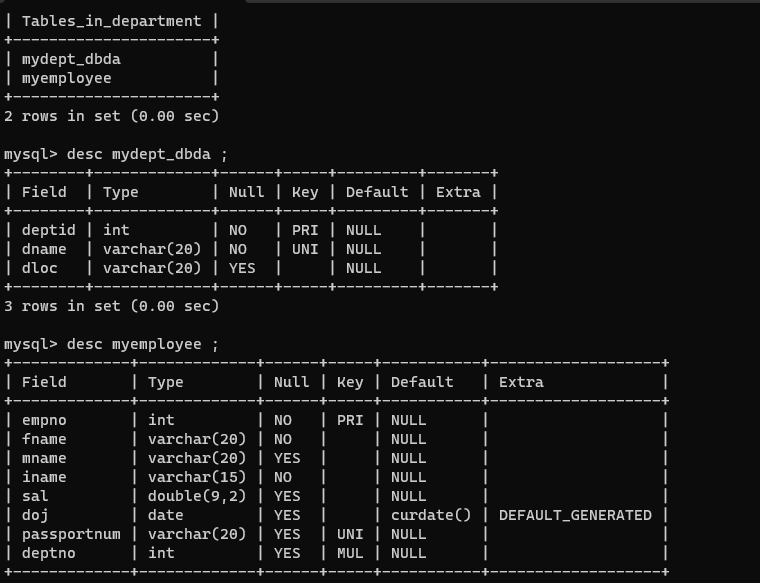
sal double(9,2) check(sal>=1000),

doj date default (current\_date),

passportnum varchar(20) unique,

deptno int(5), constraint fk\_deptno foreign key(deptno) references mydept\_DBDA(deptid) on delete cascade

)



32. Create following tables Student, Course

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

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

Marks(studid,courseid,marks)

Sample data for marks table

studid,courseid,marks

1 1 99

1 3 98

2 1 95

2 2 97

create table marks(

studid number,

courseid number,

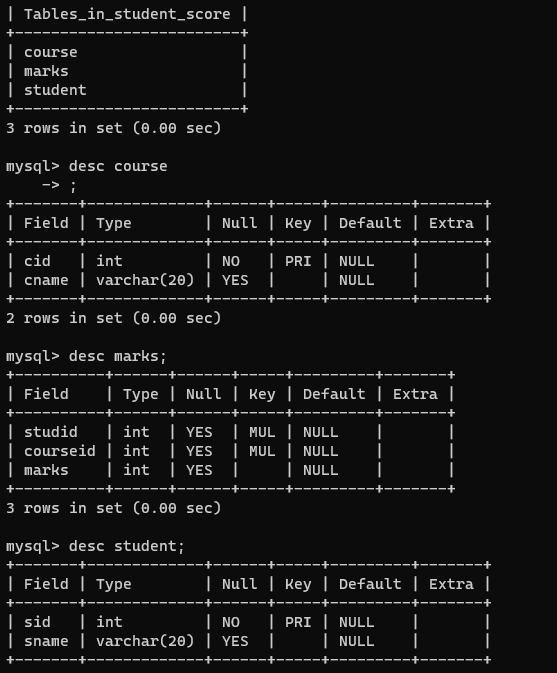
marks number,

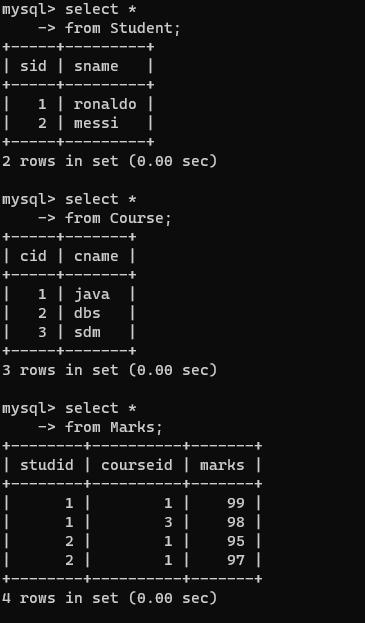
constraint pk primary key(studid,courseid),

constraint fk\_sid foreign key (studid) references student(sid) on delete cascade,

constraint fk\_cid foreign key (courseid) references course(cid)

)





33. Create empty table emp10 with table structure same as emp table.

create table emp10 as

(

select \*

from emp

where 1=2;

)

=> create view emp10

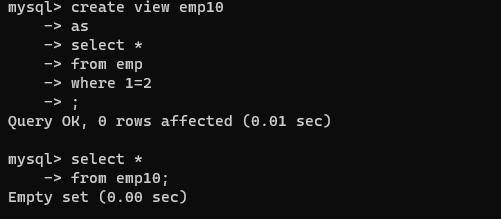
as

select \*

from emp

where 1=2

;



34. 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

=>

mysql> ALTER TABLE emp

ADD CONSTRAINT pk\_emp PRIMARY KEY (empno);

ALTER TABLE dept

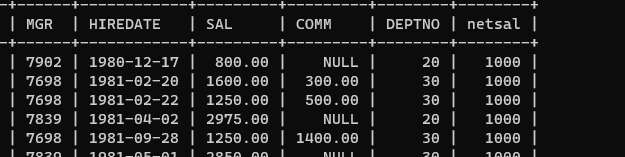
ADD CONSTRAINT pk\_dep PRIMARY KEY (deptno);

mysql> ALTER TABLE salgrade

ADD CONSTRAINT pk\_sal PRIMARY KEY (grade);

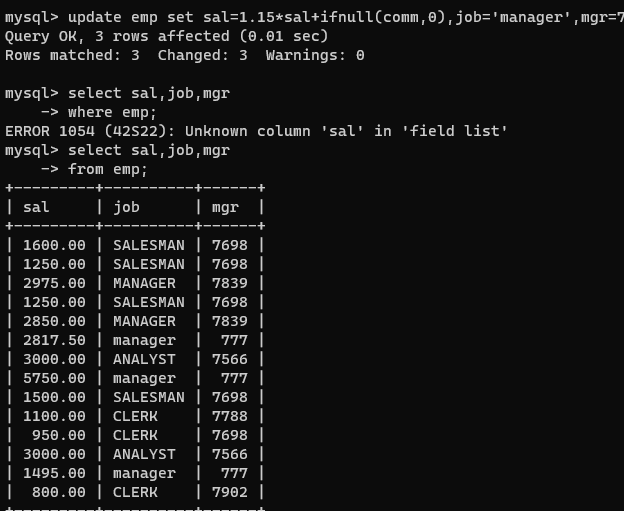


ALTER TABLE emp ADD netsal INT DEFAULT 1000 NOT NULL;



35. 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.

=> update emp set sal=1.15\*sal+ifnull(comm,0),job='manager',mgr=777 where deptno=10;



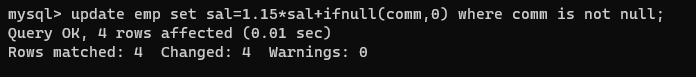
36. change job of smith to senior clerk

=> update emp set job='s.clerk' where ename='smith'



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

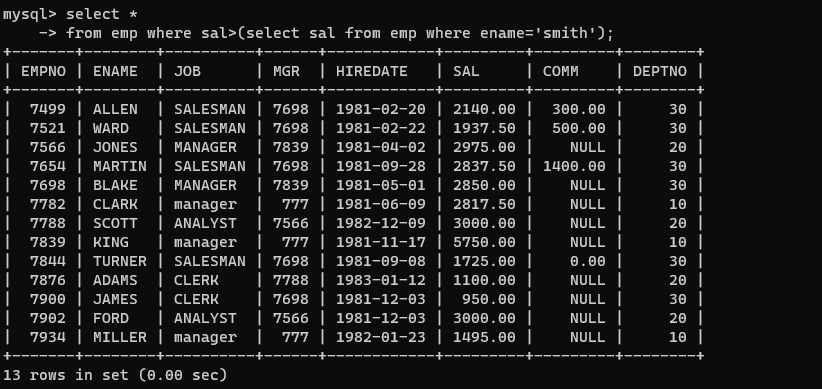
=>update emp set sal=1.15\*sal+ifnull(comm,0) where comm is not null;



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

=> select \*

from emp where sal>(select sal from emp where ename='smith')

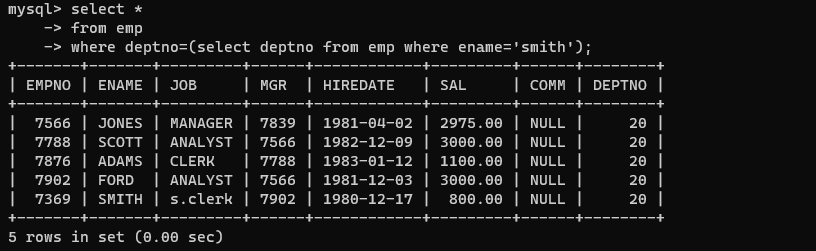


39. list all employees who are working in smith's department

=> select \*

from emp

where deptno=(select deptno from emp where ename='smith')

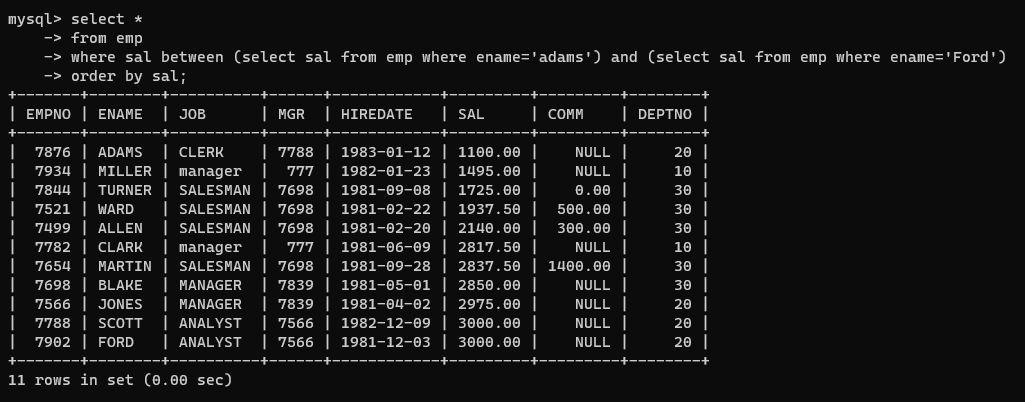


40. list all employees with sal < rajan's sal and salary > revati's sal

=>select \*

from emp

where sal between (select sal from emp where ename='adams') and (select sal from emp where ename='Ford')



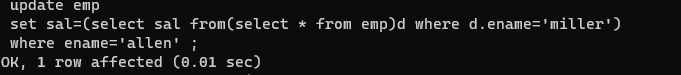
41. delete all employees working in alan's department

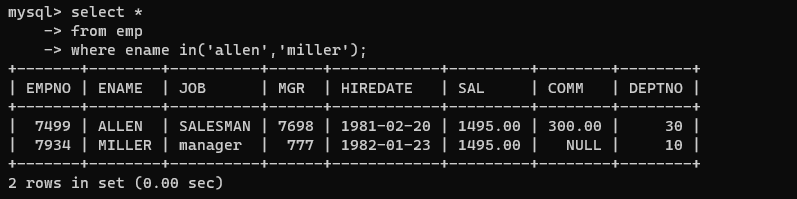
42. change salary of Alan to the salary of Miller.

update emp

set sal=(select sal from(select \* from emp)d where d.ename='miller')

where ename='allen' ;



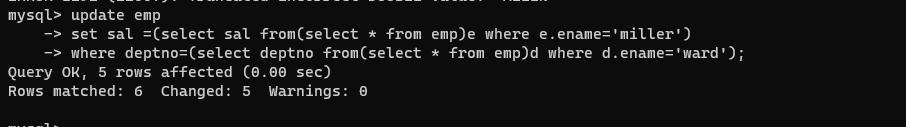


43. change the salary of all employees who work in Wall's department to the salary of Miller.

update emp

set sal =(select sal from(select \* from emp)e where e.ename='miller')

where deptno=(select deptno from(select \* from emp)d where d.ename='ward');

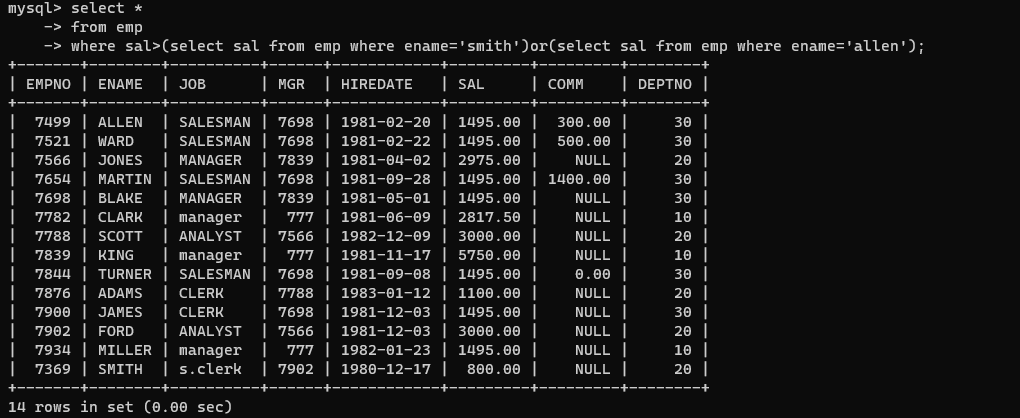


44. list all employees with salary > either Smith's salary or alan's sal

select \*

from emp

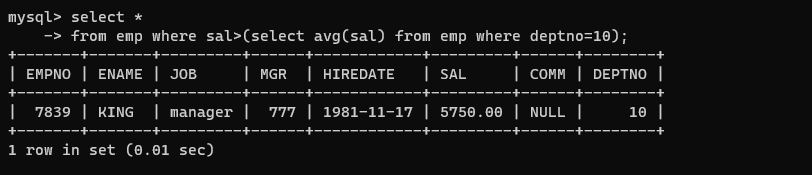
where sal>(select sal from emp where ename='smith')or(select sal from emp where ename='allen');



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

select \*

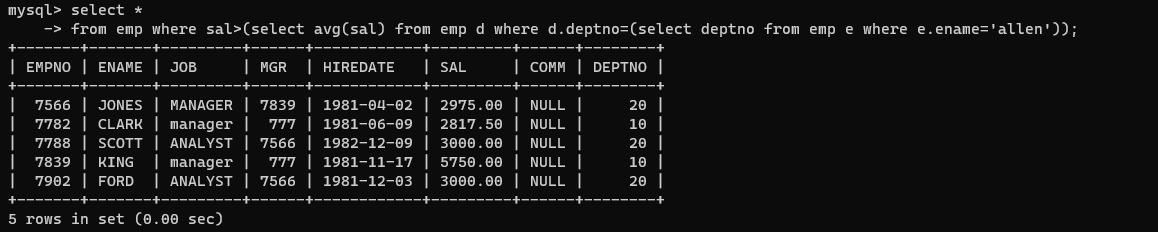
from emp where sal>(select avg(sal) from emp where deptno=10);



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

select \*

from emp where sal>(select avg(sal) from emp d where d.deptno=(select deptno from emp e where e.ename='allen'));

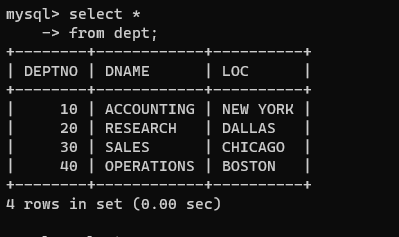


47. list all employees who are working in purchase department

select \*

from emp

where deptno=(select deptno from dept where dname='purchase');

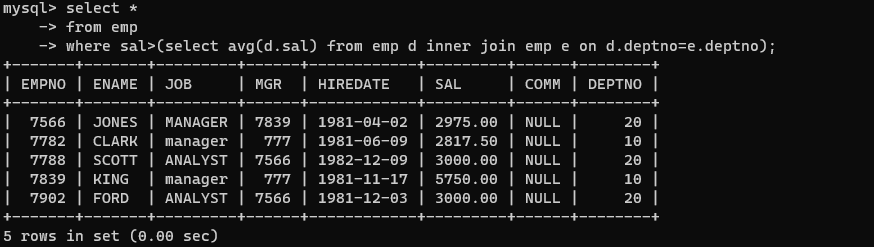


48. list all employees who earn more than average salary of their own department

select \*

from emp

where sal>(select avg(d.sal) from emp d inner join emp e on d.deptno=e.deptno);

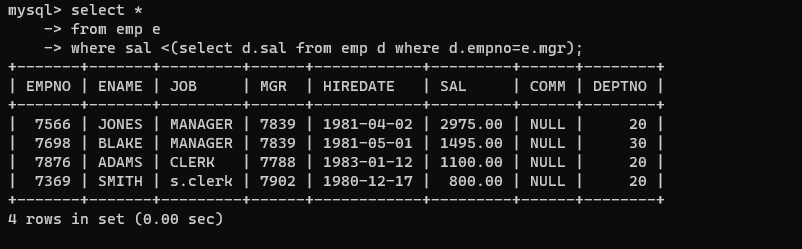


49. list all employees who earn sal < than their managers salary

select \*

from emp e

where sal <(select d.sal from emp d where d.empno=e.mgr);



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

select \*

-> from emp

-> where sal>(select avg(e.sal) from emp e inner join emp d on e.job=d.job);



51. display employee name and department

select e.ename,d.dname

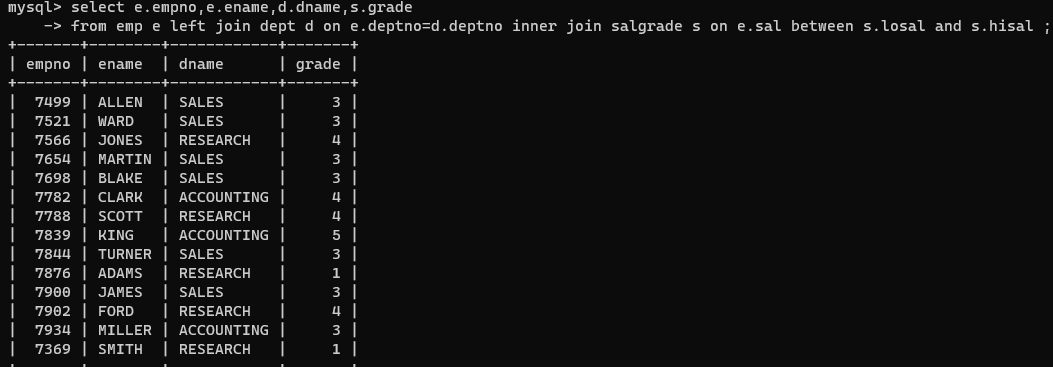
from emp e left join dept d on e.deptno=d.deptno;



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

select e.empno,e.ename,d.dname,s.grade

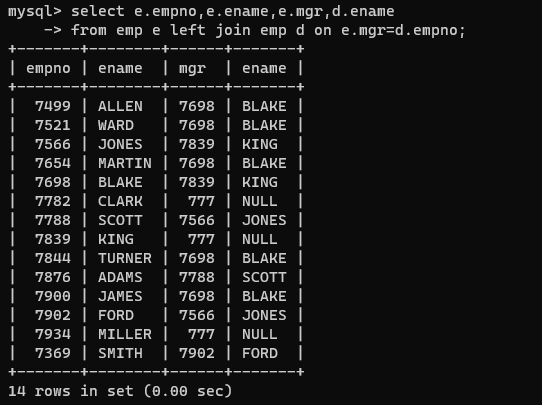
from emp e left join dept d on e.deptno=d.deptno inner join salgrade s on e.sal between s.losal and s.hisal ;



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

select e.empno,e.ename,e.mgr,d.ename

from emp e left join emp d on e.mgr=d.empno;



54. 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)

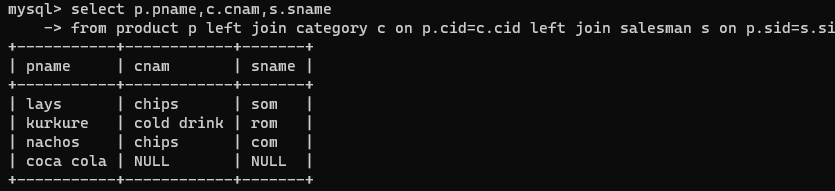
salesman (sid,sname,address)

category(cid,cnam,descritpion)

1. list all product name,their category name and name of a person, who sold that product

select p.pname,c.cnam,s.sname

from product p left join category c on p.cid=c.cid left join salesman s on p.sid=s.sid;

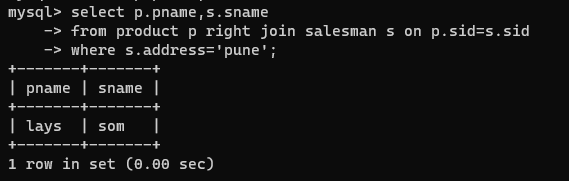


2. list all product name and salesman name for all salesman who stays in pune

select p.pname,s.sname

from product p right join salesman s on p.sid=s.sid

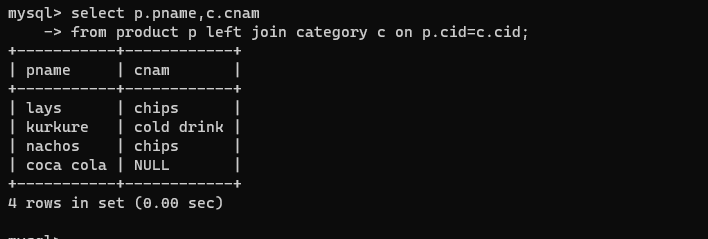
where s.address='pune';



3. list all product name and category name

select p.pname,c.cnam

from product p left join category c on p.cid=c.cid;



55. create following tables and solve following questions(primary keys are marked in yellow)

foreign keys are marked in green

faculty(fid,fname,sp.skill1,sp.skill2)

courses(cid,cname,rid,fid)

room(roomid,rname,rloc)

faculty

fid fname spskill1 spskill2

10 kjzhcjhz a b

11 sdd x z

12 lksjk a x

13 ksdjlkj a b

courses

cid cname rid fid

121 DBDA 100 10

131 DAC 101

141 DTISS

151 DIOT 105 12

Room

roomid rname rloc

100 jasmin 1st floor

101 Rose 2nd floor

105 Lotus 1st floor

103 Mogra 1st floor

1. list all courses for which no room is assigned and all rooms for which are

Available

SELECT c.cid, c.cname,null

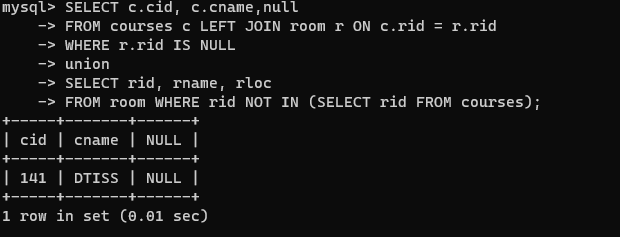
FROM courses c LEFT JOIN room r ON c.rid = r.rid

WHERE r.rid IS NULL

union

SELECT rid, rname, rloc

FROM room WHERE rid NOT IN (SELECT rid FROM courses);



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

allocated to any course

select f.fid,f.fname

from faculty f left join courses c on f.fid=c.fid

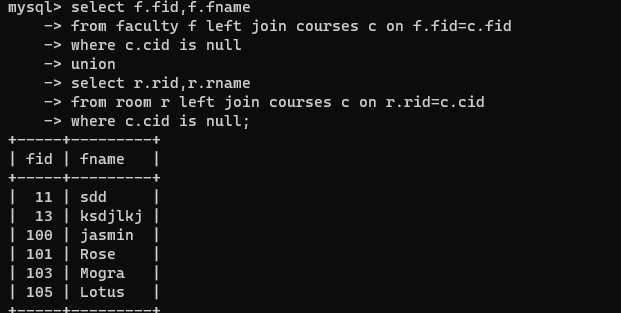
where c.cid is null

union

select r.rid,r.rname

from room r left join courses c on r.rid=c.cid

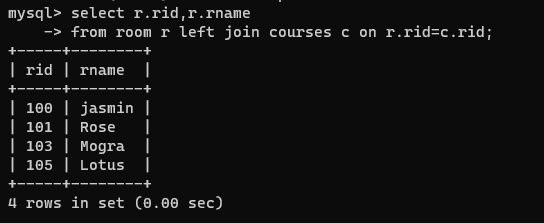
where c.cid is null;



3. list all rooms which are allocated or not allocated to any courses

select r.rid,r.rname

from room r left join courses c on r.rid=c.rid;

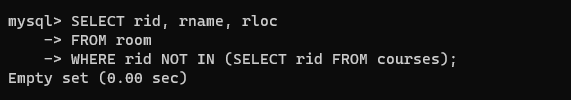


4. list all rooms which are not allocated to any courses

SELECT rid, rname, rloc

FROM room

WHERE rid NOT IN (SELECT rid FROM courses);



5. display courses and faculty assigned to those courses whose special skill is

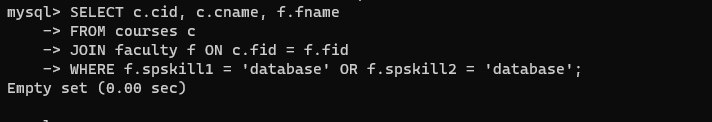
Database

SELECT c.cid, c.cname, f.fname

FROM courses c

JOIN faculty f ON c.fid = f.fid

WHERE f.spskill1 = 'database' OR f.spskill2 = 'database';



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

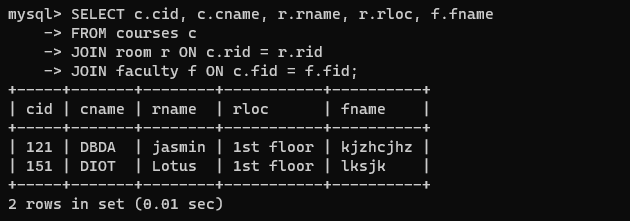
Details

SELECT c.cid, c.cname, r.rname, r.rloc, f.fname

FROM courses c

JOIN room r ON c.rid = r.rid

JOIN faculty f ON c.fid = f.fid;



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

saleman ----- sname -----not null

sid sname city

11 Rahul Pune

12 Kirti Mumbai

13 Prasad Nashik

14 Arnav Amaravati

category ---- cname unique and not null

cid cname description

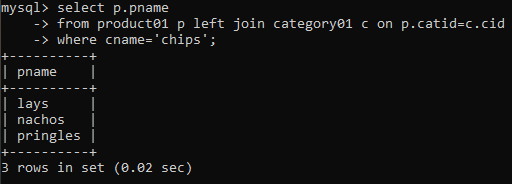
1 chips very crunchy

2 chocolate very chocolaty

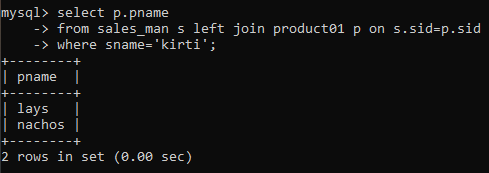
3 snacks yummy

4 cold drinks thanda thanda cool cool

1. List all products with category chips



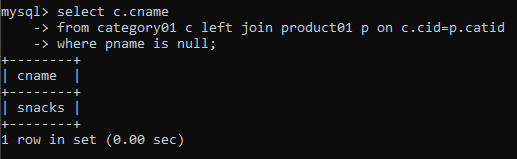
1. display all products sold by kirti



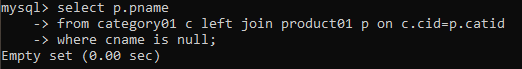
1. display all salesman who do not sold any product



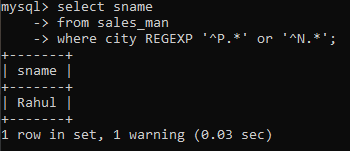
1. display all category for which no product is there



1. display all products with no category assigned



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



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



**Assignment 3**

**Indexes\_Assignments**

1. create all given tables

=> create table vehicle (

vid int primary key,

vname varchar(20),

price double(9,2),

descp varchar(100)

);

create table customer (

custid int primary key,

cname varchar(20),

address varchar(30)

);

create table salesman (

sid int primary key,

sname varchar(20),

address varchar(30)

);

create table cust\_vehicle (

custid int,

vid int,

sid int,

Buy\_price double(9,2),

primary key(custid,vid),

foreign key (custid) references customer(custid),

foreign key (vid) references vehicle(vid),

foreign key (sid) references salesman(sid)

);

insert into vehicle values(1,'Activa',80000,'ksldfjksj');

insert into vehicle values(2,'Santro',800000,'kdjfkjsd');

insert into vehicle values(3,'Motor bike',100000,'fdkdf');

insert into customer values (1,'nilima','pimpari');

insert into customer values (2,'ganesh','pune');

insert into customer values (3,'pankaj','mumbai');

insert into salesman values(10,'Rajesh','mumbai');

insert into salesman values(11,'Seema','pune');

insert into salesman values(13,'Rakhi','pune');

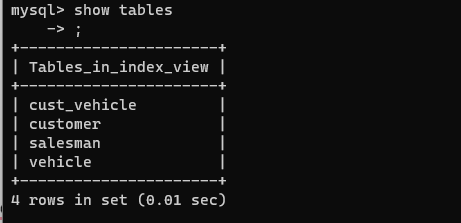
insert into cust\_vehicle values (1,1,10,75000);

insert into cust\_vehicle values (1,2,10,790000);

insert into cust\_vehicle values (2,3,11,80000);

insert into cust\_vehicle values (3,3,11,75000);

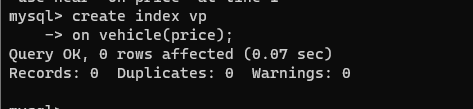
insert into cust\_vehicle values (3,2,10,800000);



2. create index on vehicle table based on price

=> create index vp

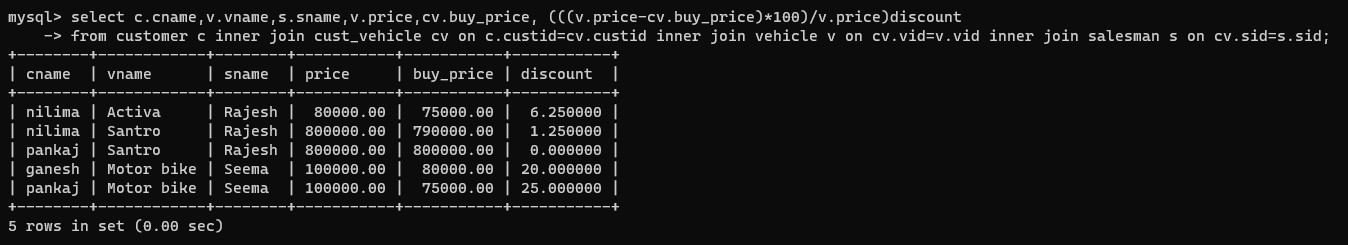
-> on vehicle(price);



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

=> select c.cname,v.vname,s.sname,v.price,cv.buy\_price, (((v.price-cv.buyprice)\*100)/v.price)discount

from customer c inner join cust\_vehicle cv on c.custid=cv.custid inner join vehicle v on cv.vid=v.vid inner join salesman s on cv.sid=s.sid;

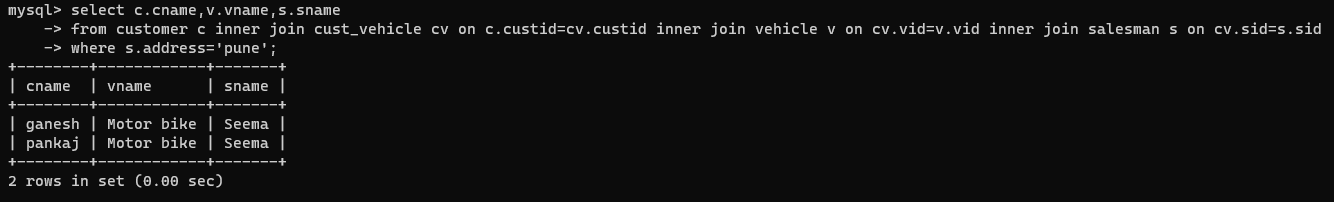


4. find all customer name,vehicle name,salesman name for all salesman who stays in pune

=> select c.cname,v.vname,s.sname

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

where s.address='pune';

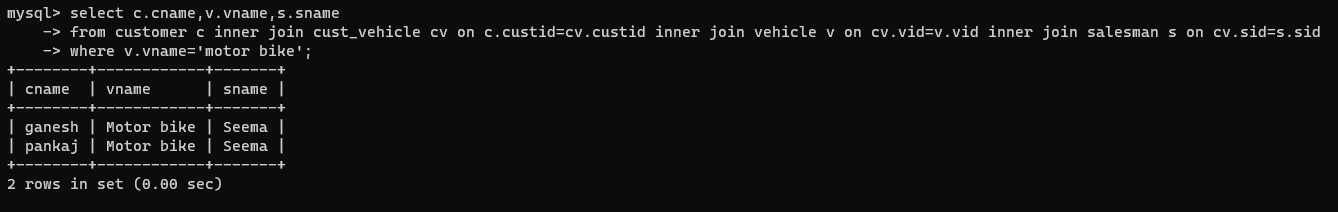


5. find how many customers bought motor bike

=> select c.cname,v.vname,s.sname

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

where v.vname='motor bike';



6. create a view find\_discount which displays output

-------to create view

=>create view find\_discount

as

select cname,vname,price,buy\_price,(price-buy\_price) 'discount'

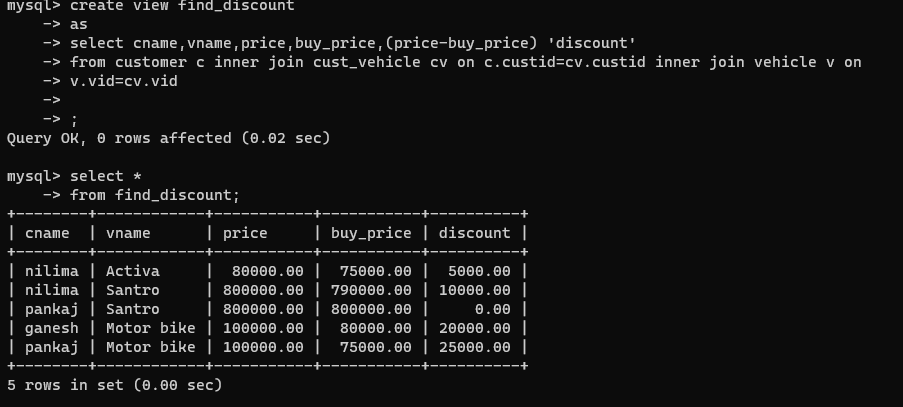
from customer c inner join cust\_vehicle cv on c.custid=cv.custid inner join vehicle v on

v.vid=cv.vid

;

--------to display discount

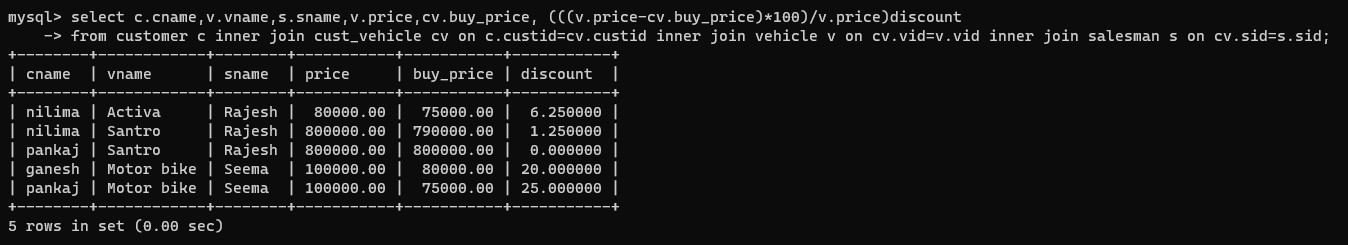
select \* from find\_discount;



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

=> select c.cname,v.vname,s.sname,v.price,cv.buy\_price, (((v.price-cv.buyprice)\*100)/v.price)discount

from customer c inner join cust\_vehicle cv on c.custid=cv.custid inner join vehicle v on cv.vid=v.vid inner join salesman s on cv.sid=s.sid;



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

Commission

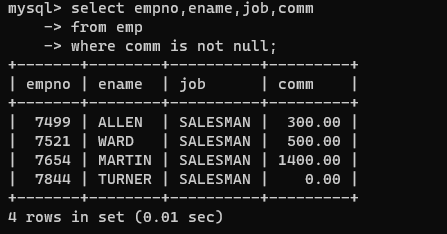
=> create view my\_hr

AS

select empno,ename,job,comm

from emp

where comm is not null;



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

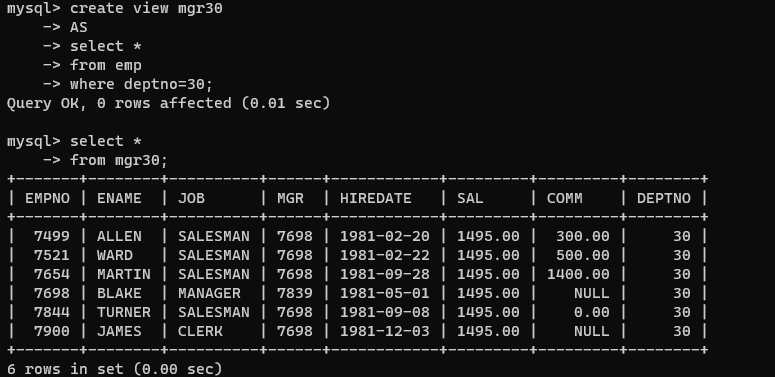
=>create view mgr30

AS

select \*

from emp

where deptno=30;

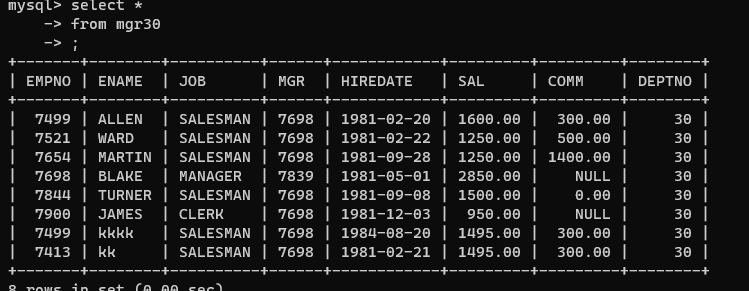


10. insert 3 employees in view mgr30 check whether insertion is possible

=> insert into mgr30 values(7413,'kk','SALESMAN',7698, '1981-02-21',1495.00,300.00,30);

insert into mgr30 values(7494,'kkk','SALESMAN',7698, '1981-05-20',1495.00,300.00,20);

insert into mgr30 values(7499,'kkkk','SALESMAN',7698, '1984-08-20',1495.00,300.00,30);

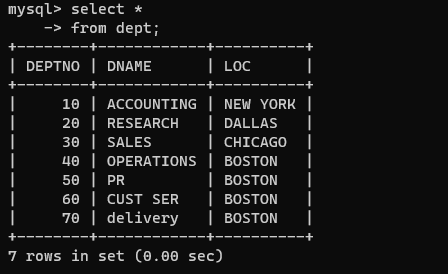


11. insert 3 records in dept and display all records from dept

=>insert into dept values (50,'PR','BOSTON');

insert into dept values (60,'CUST SER','BOSTON');

insert into dept values (70,'delivery','BOSTON');



12. use rollback command check what happens

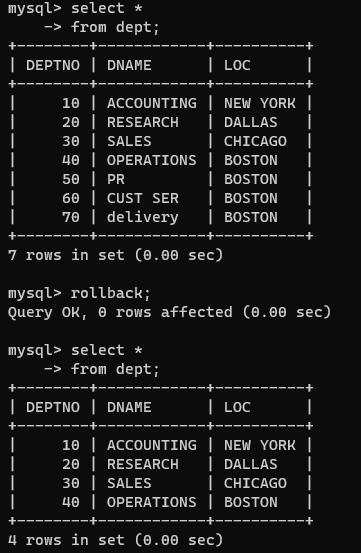
=>start transaction;

insert into dept values (50,'PR','BOSTON');

insert into dept values (60,'CUST SER','BOSTON');

insert into dept values (70,'delivery','BOSTON');

rollback;



13. 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

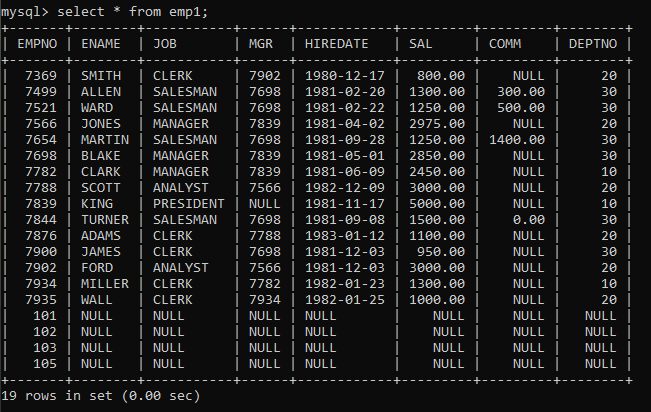
rollback upto svaepoint B

check what all records will appear in employee table

rollback upto A

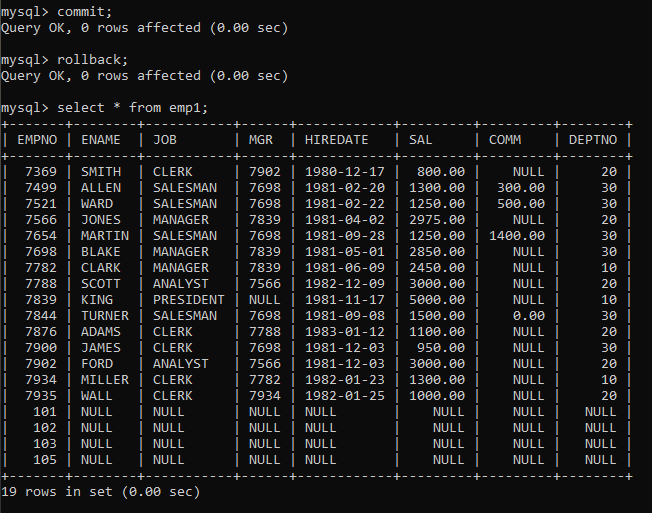
check what all records will appear in employee table

commit all changes



check what all records will appear in employee table

check whether you can roll back the contents.



14. create a procedure getMin(deptno,minsal) to find minimum salary of given table.

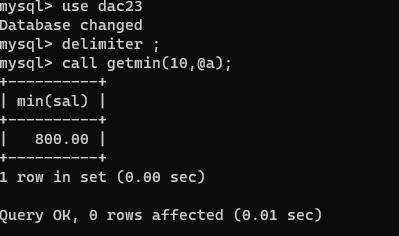
=> delimiter //

create procedure getMin(in deptno int,out minsal double(9,2))

begin

select min(sal) from emp;

end //



**Assignment 4**

**pl-sql**

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

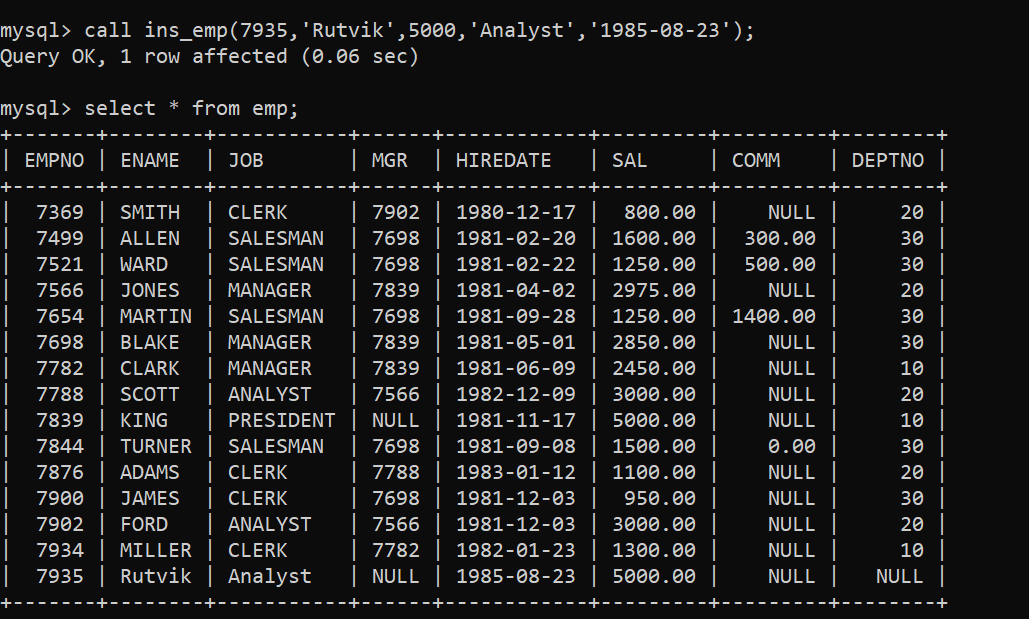
=>delimiter //

create procedure records(eno int,en varchar (20),s double(9,2),j varchar(20),hd date)

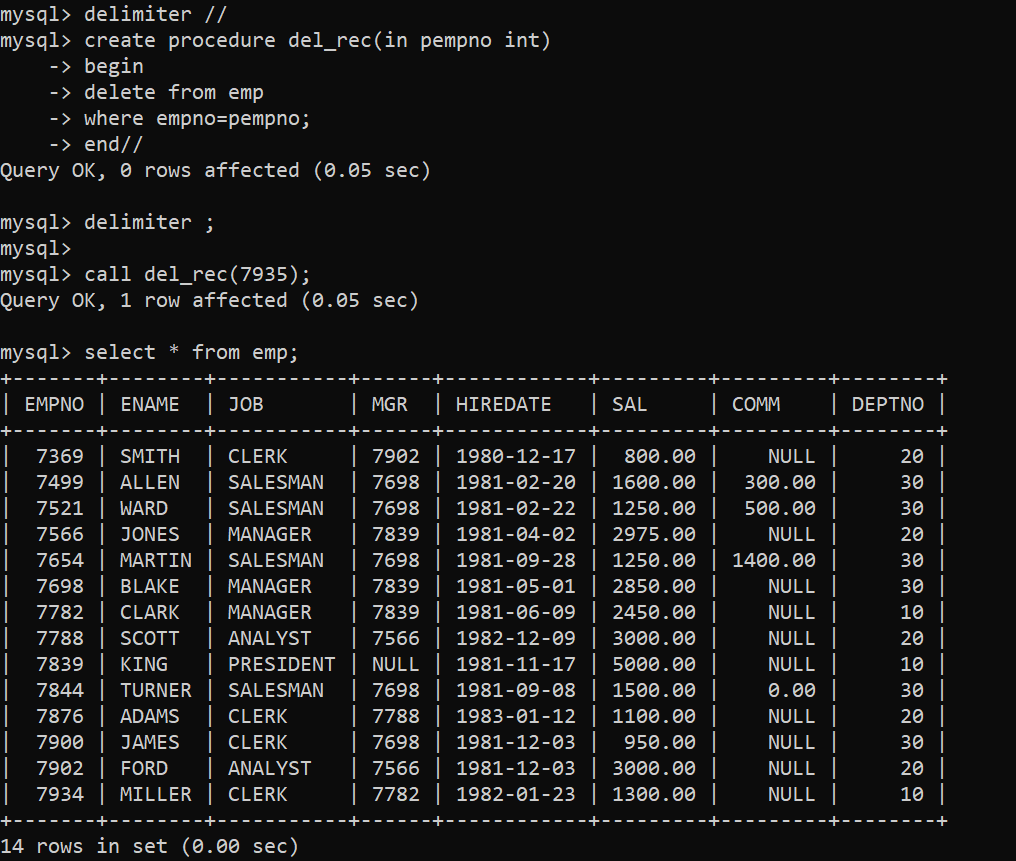
begin

insert into emp(empno,ename,sal,job,hiredate) values(eno,en,s,j,hd);

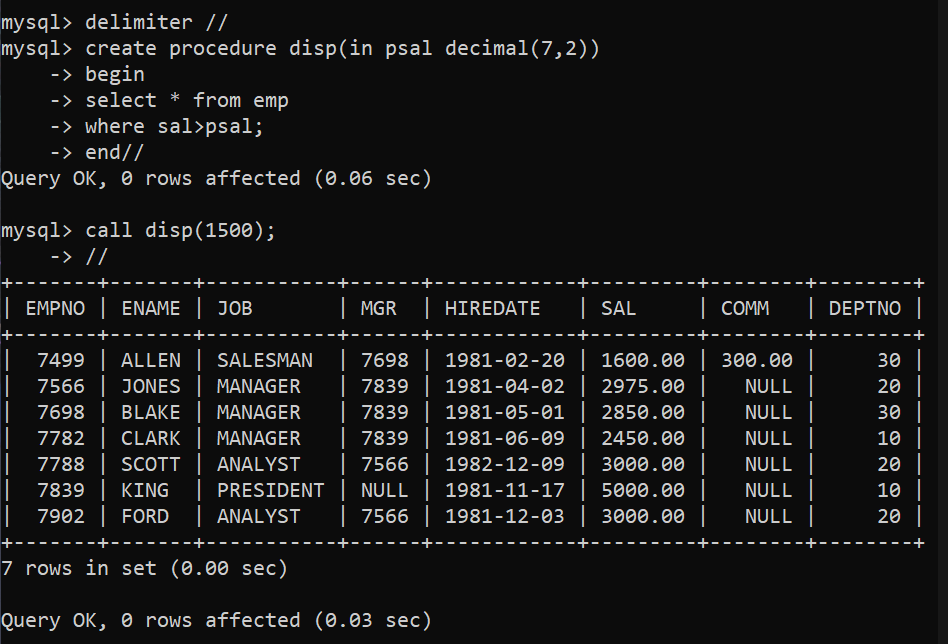
end //



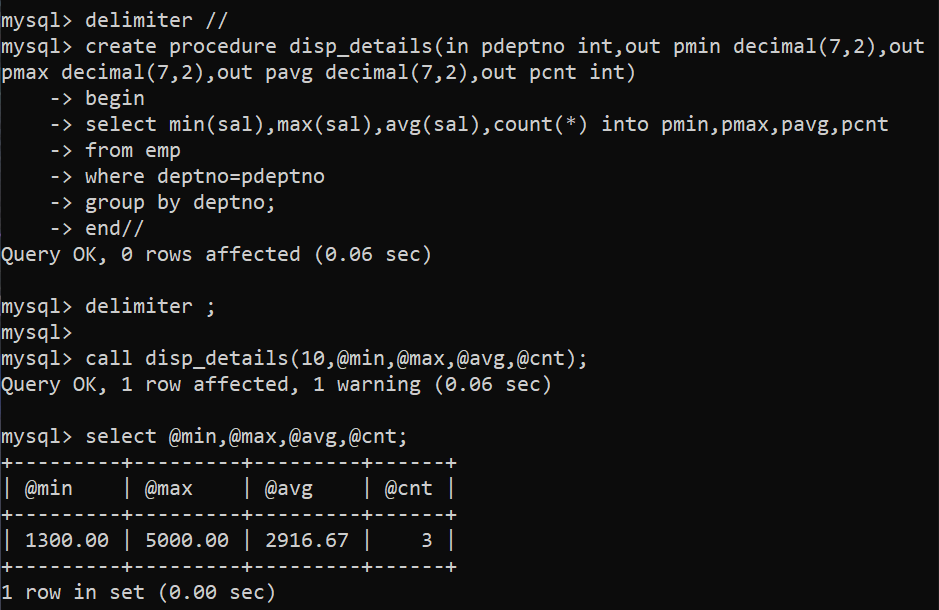
2) write a procedure to delete record from employee table. the procedure should accept empno as input parameter. write delete statement inside procedure delete\_emp to delete one record from emp table



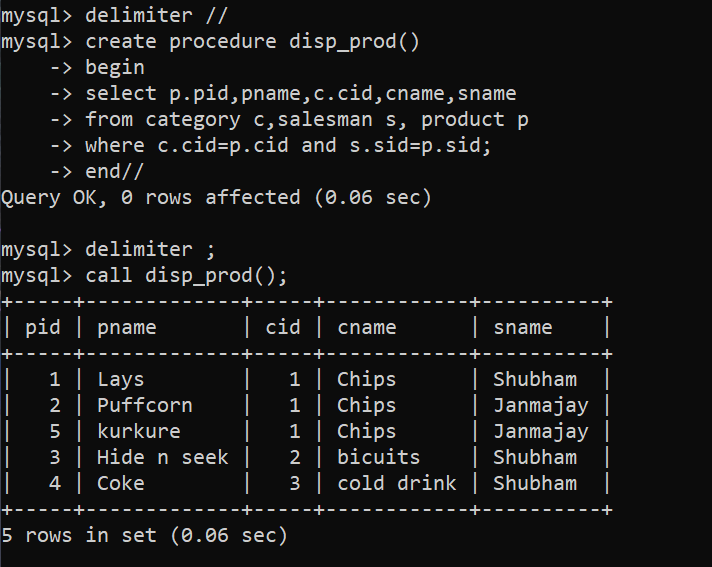
3) write a procedure to display empno,ename,deptno,dname for all employees with sal > given salary. pass salary as a parameter to procedure

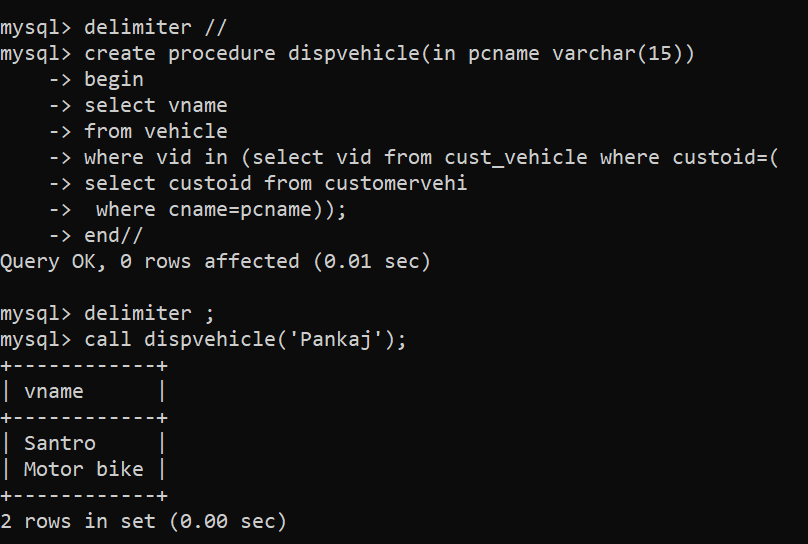


4) write a procedure to find min,max,avg of salary and number of employees in the given deptno.

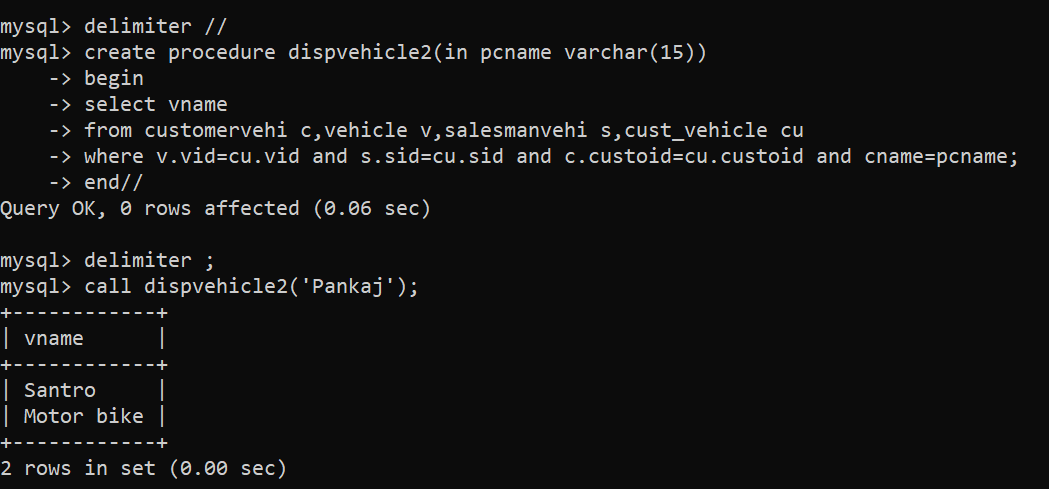


5) write a procedure to display all pid,pname,cid,cname and salesman name(use product,category and salesman table)

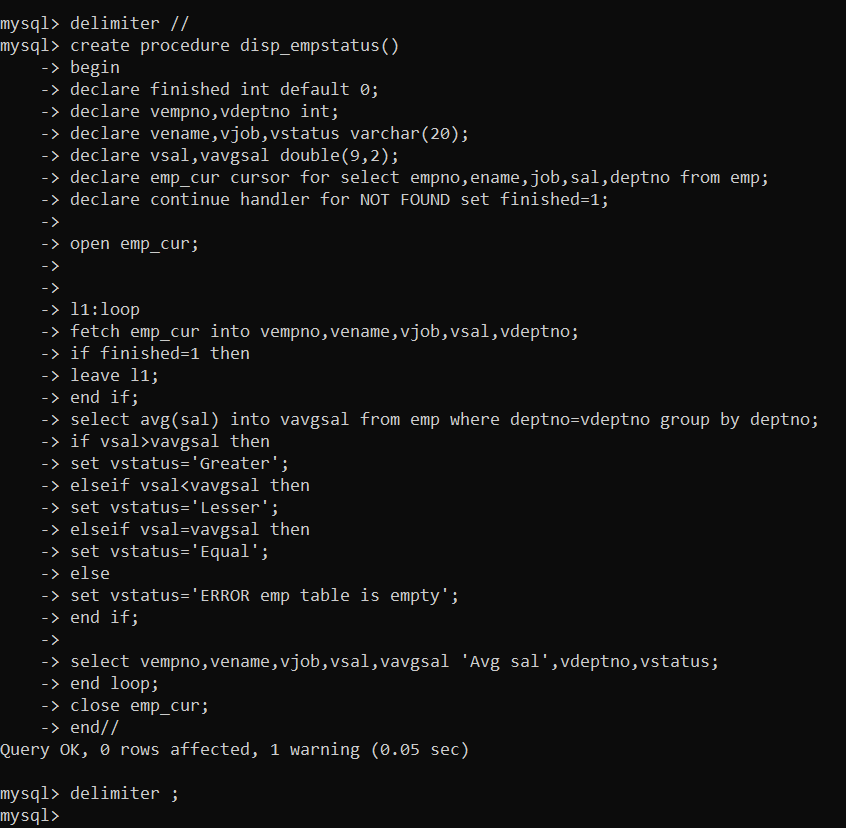


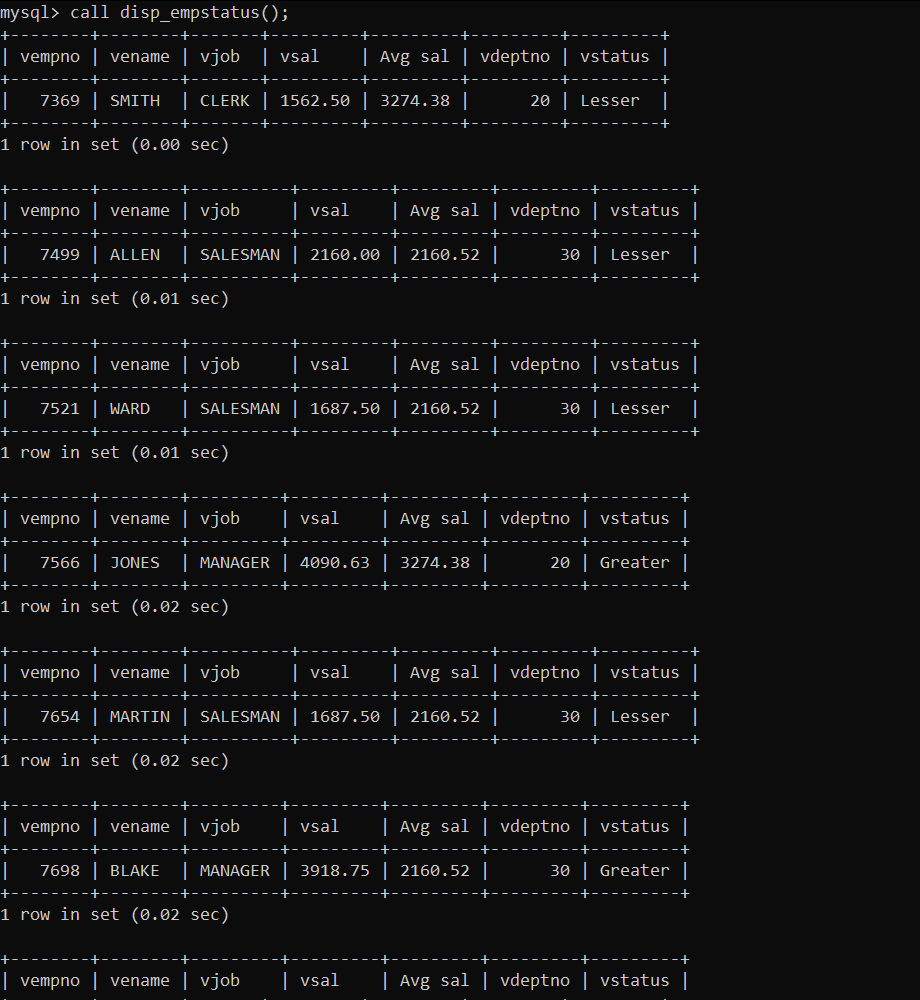
6) write a procedure to display all vehicles bought by a customer. pass cutome name as a parameter.(use vehicle,salesman,custome and relation table) 

Way 2

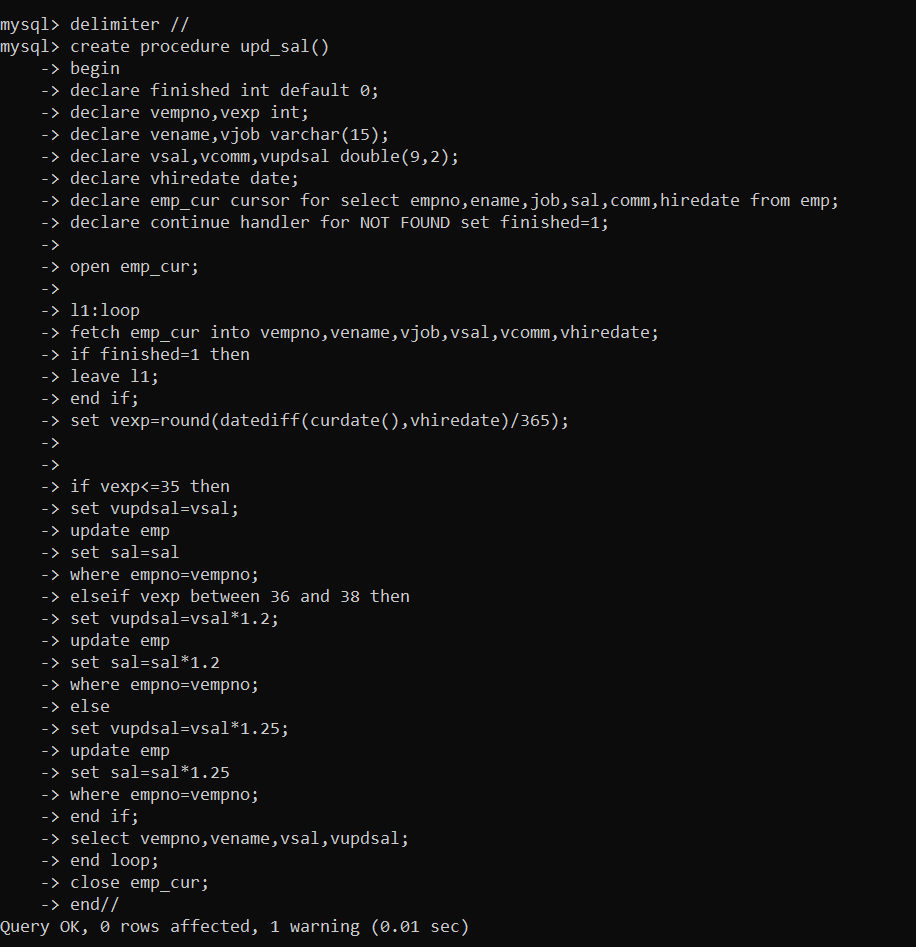


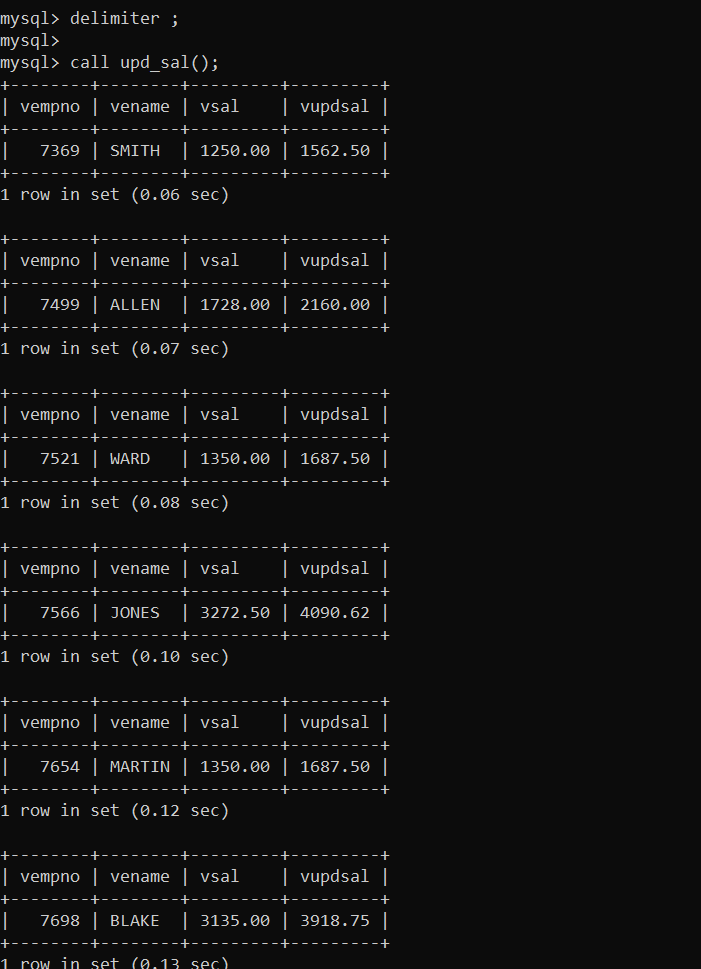
7) Write a procedure that displays the following information of all emp Empno,Name,job,Salary,Status,deptno Note: - Status will be (Greater, Lesser or Equal) respective to average salary of their own department. Display an error message Emp table is empty if there is no matching record



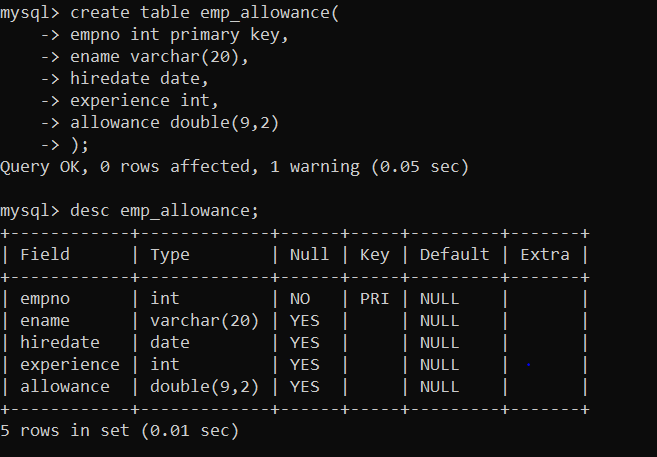


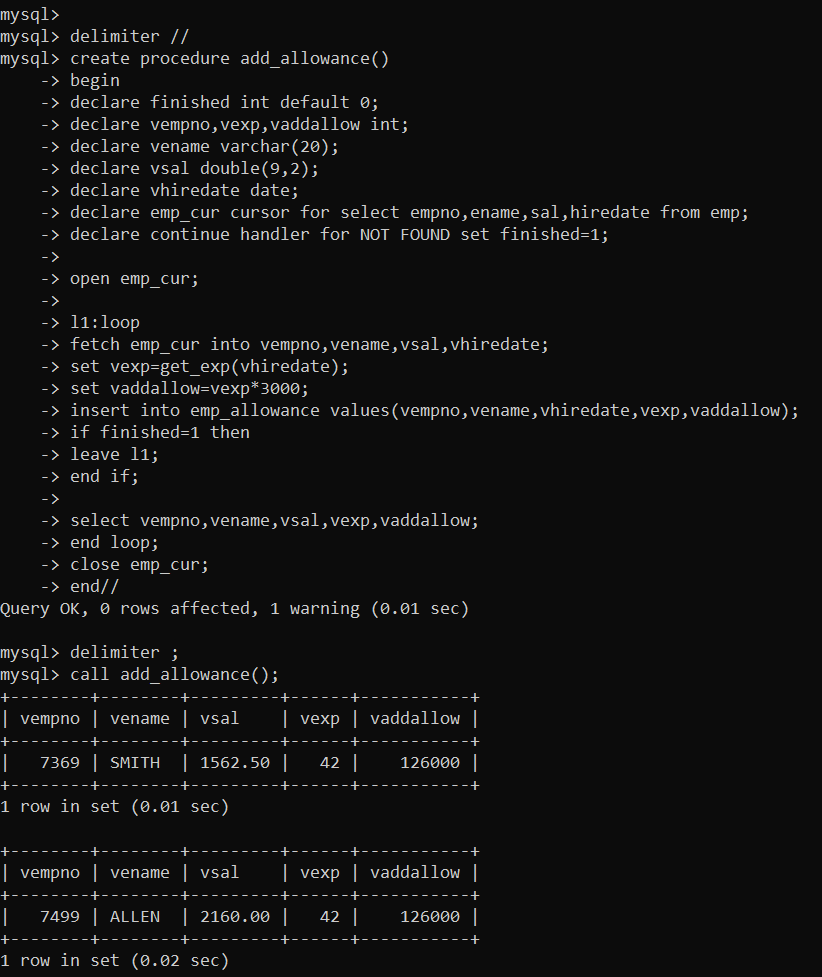
8) Write a procedure to update salary in emp table based on following rules. Exp< =35 then no Update Exp> 35 and <=38 then 20% of salary Exp> 38 then 25% of salary

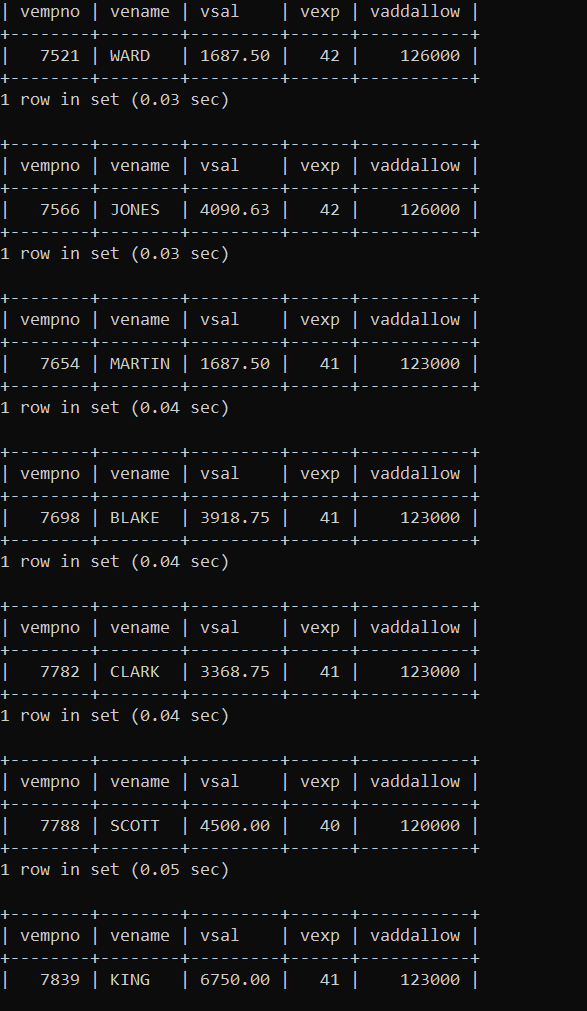


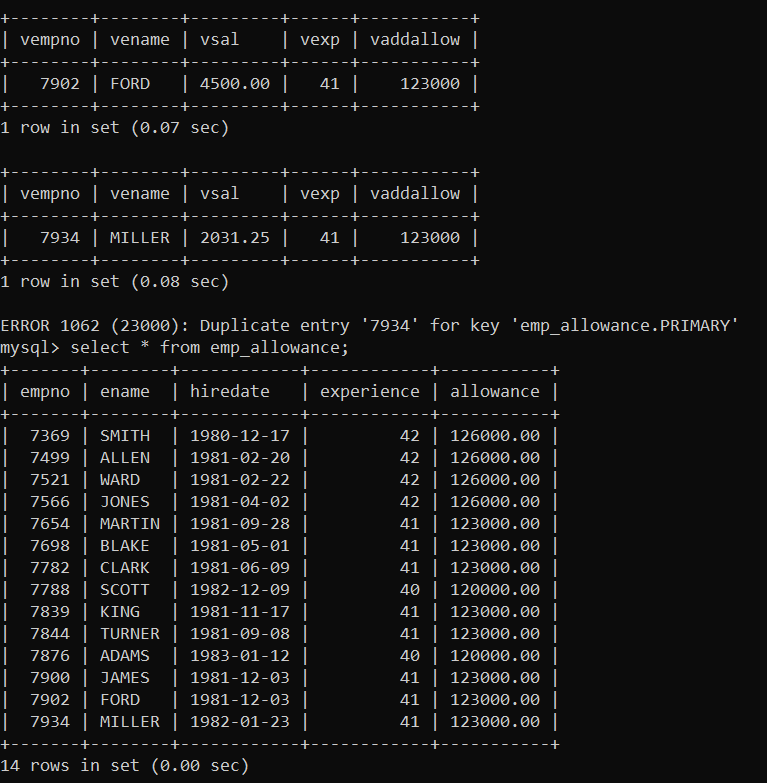


9) Write a procedure and a function. Function: write a function to calculate number of years of experience of employee.(note: pass hiredate as a parameter)

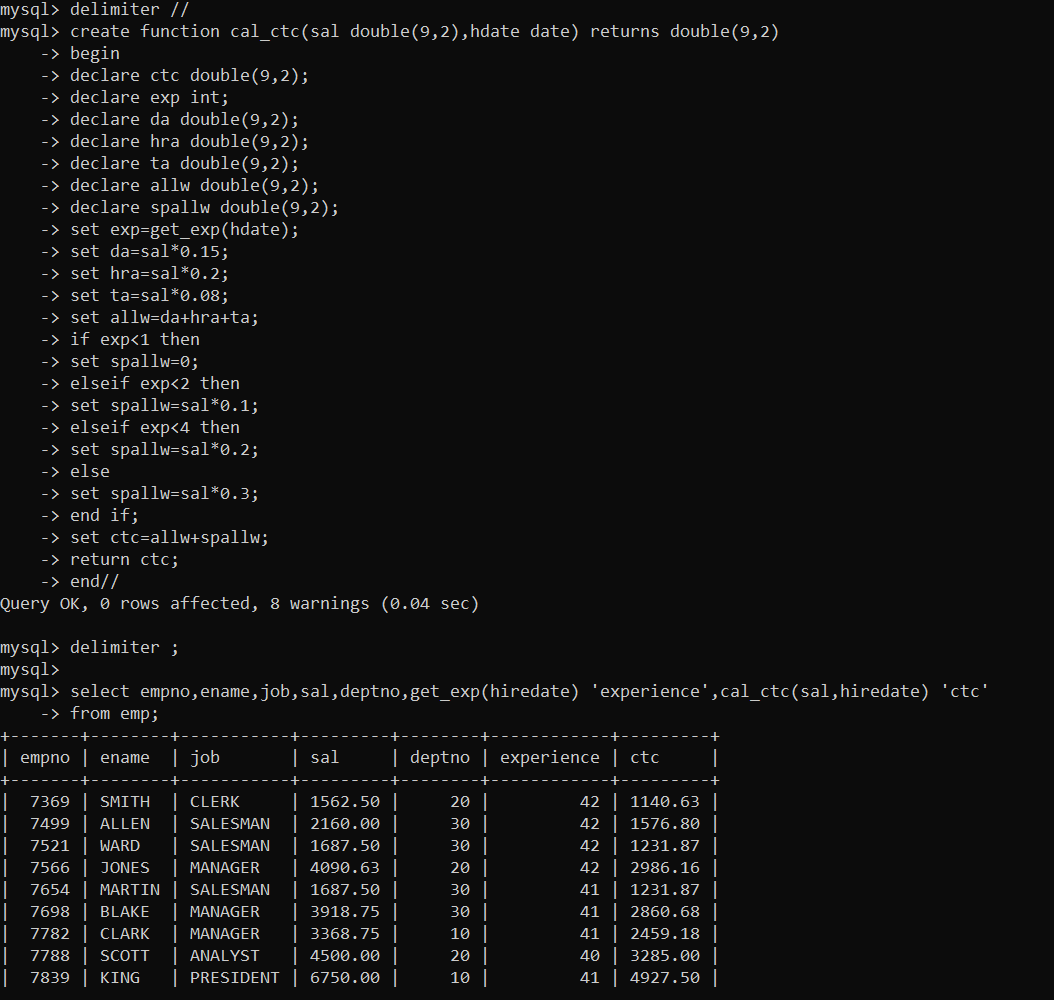




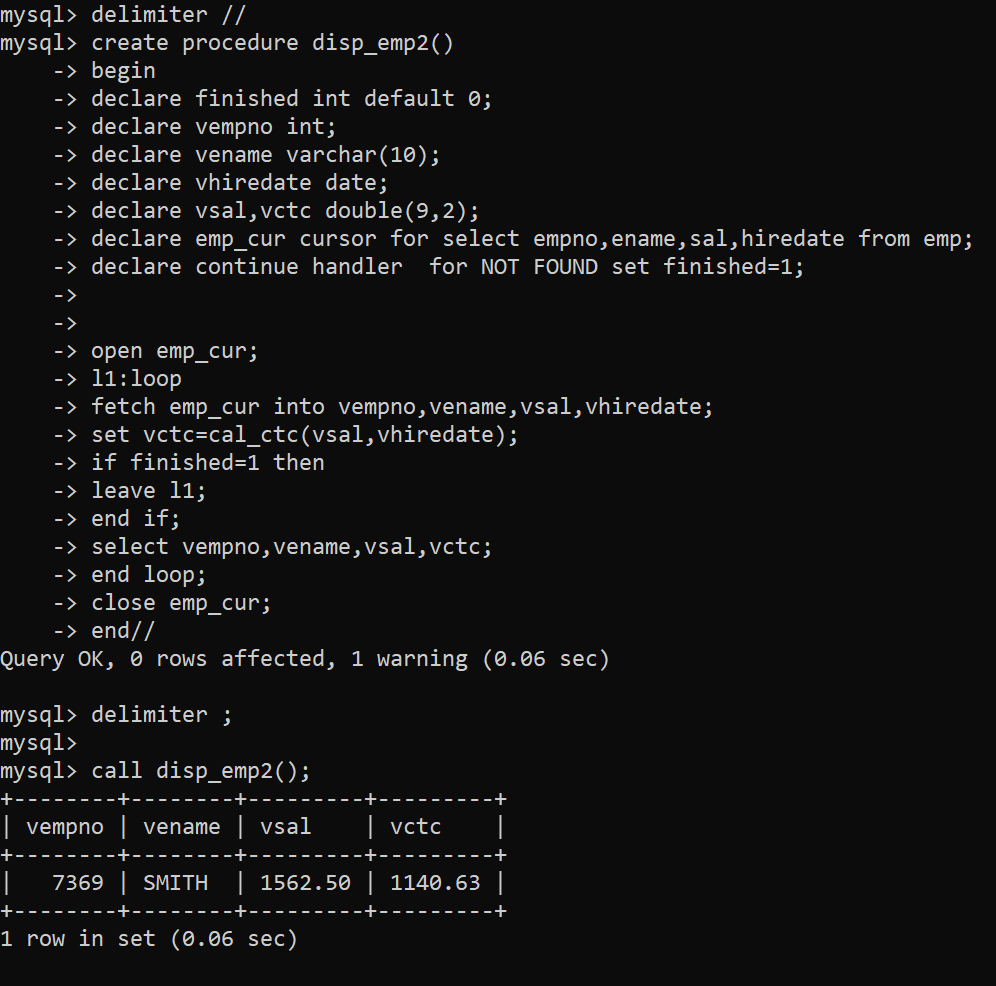


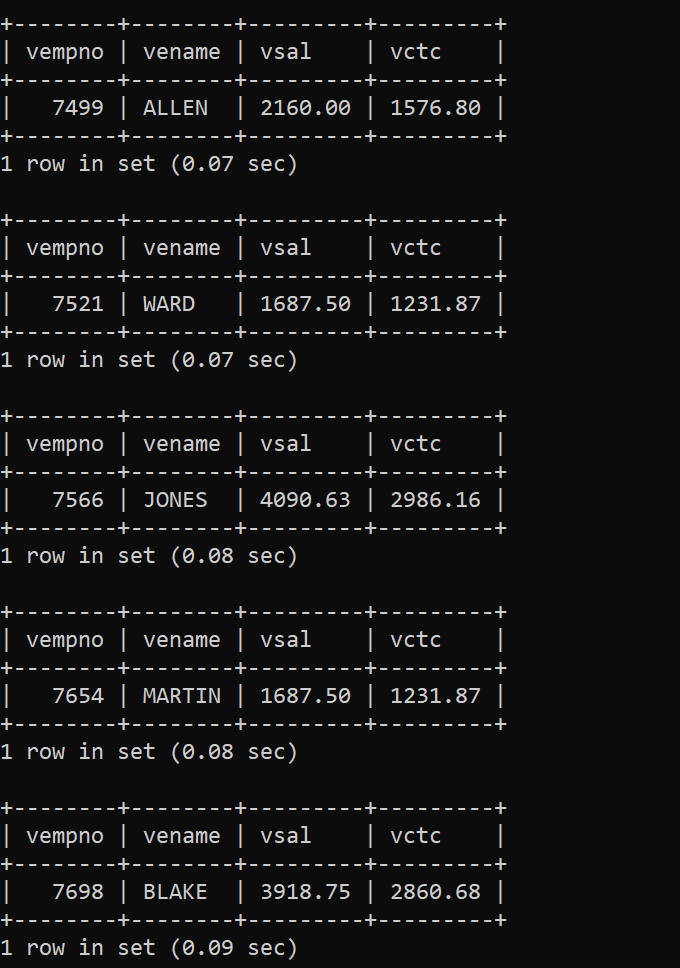


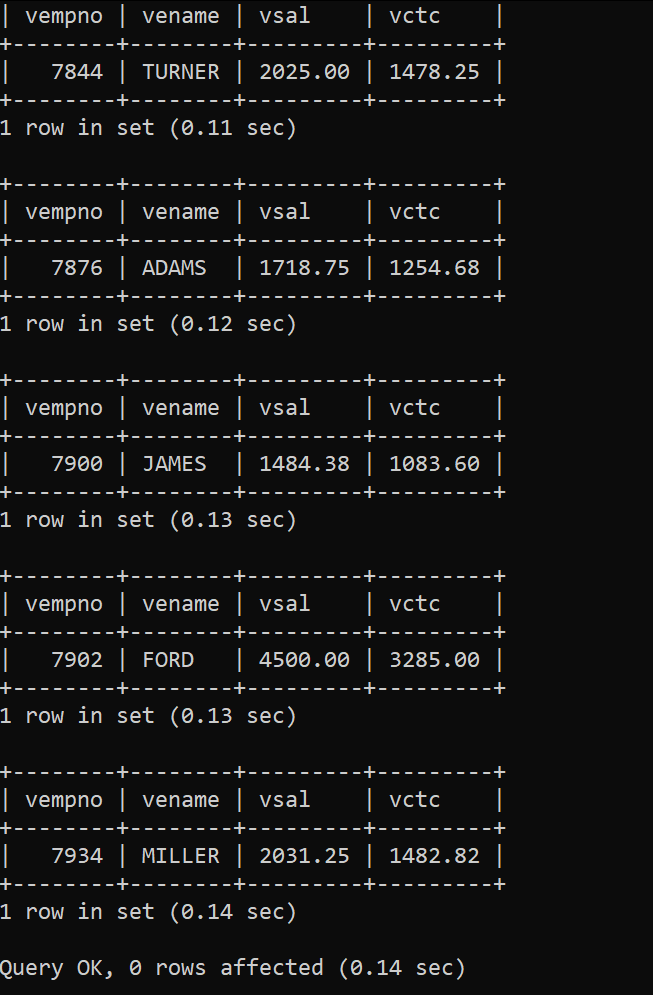
10) Write a function to compute the following. Function should take sal and hiredate as i/p and return the cost to company. DA = 15% Salary, HRA= 20% of Salary, TA= 8% of Salary. Special Allowance will be decided based on the service in the company. < 1 Year Nil >=1 Year< 2 Year 10% of Salary >=2 Year< 4 Year 20% of Salary >4 Year 30% of Salary



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

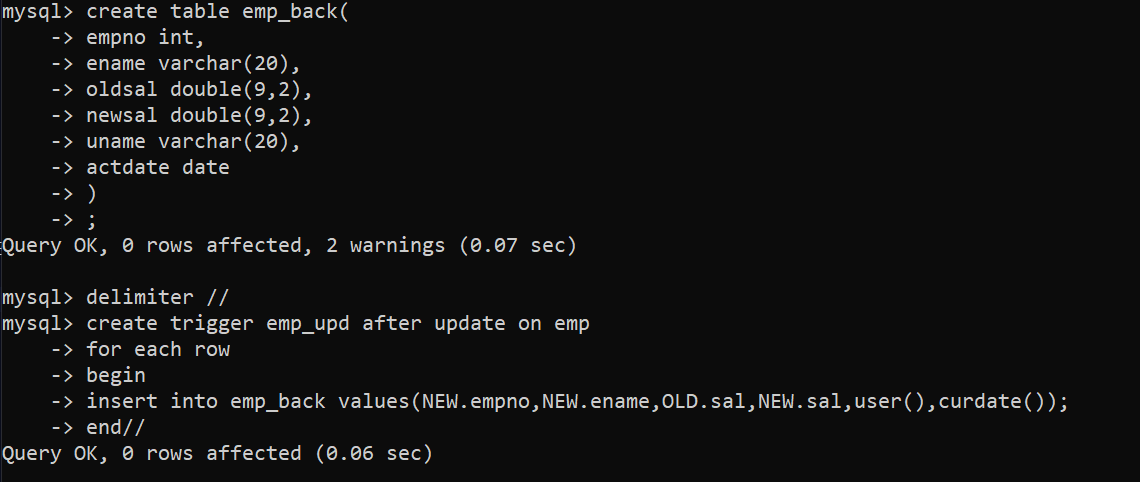


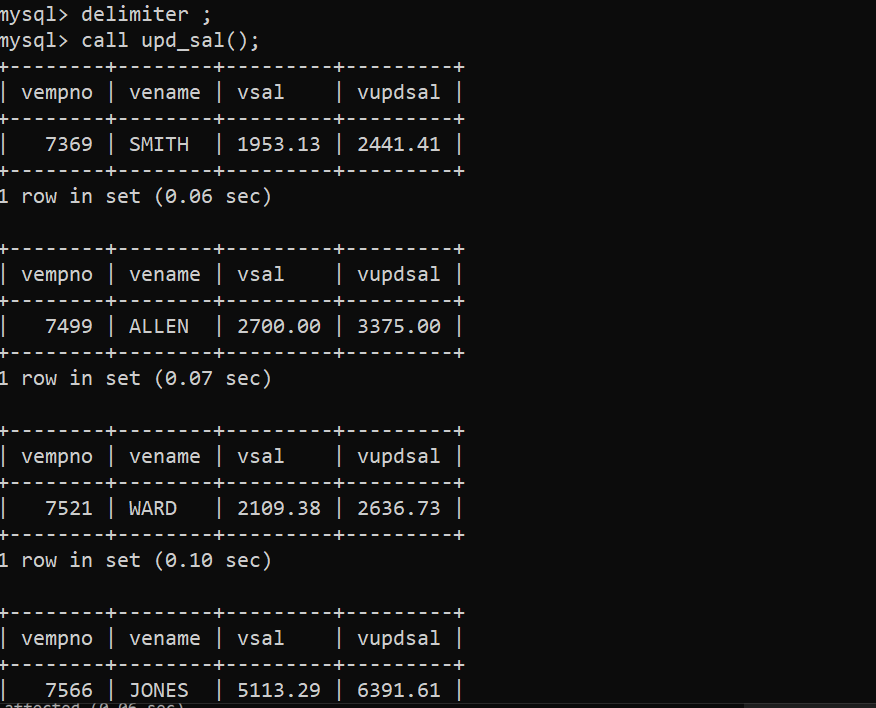


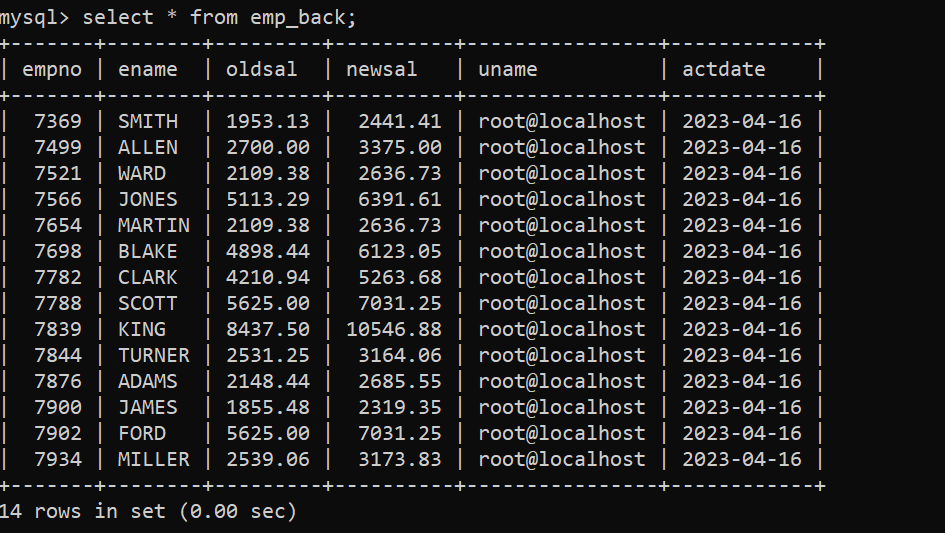


Q2)

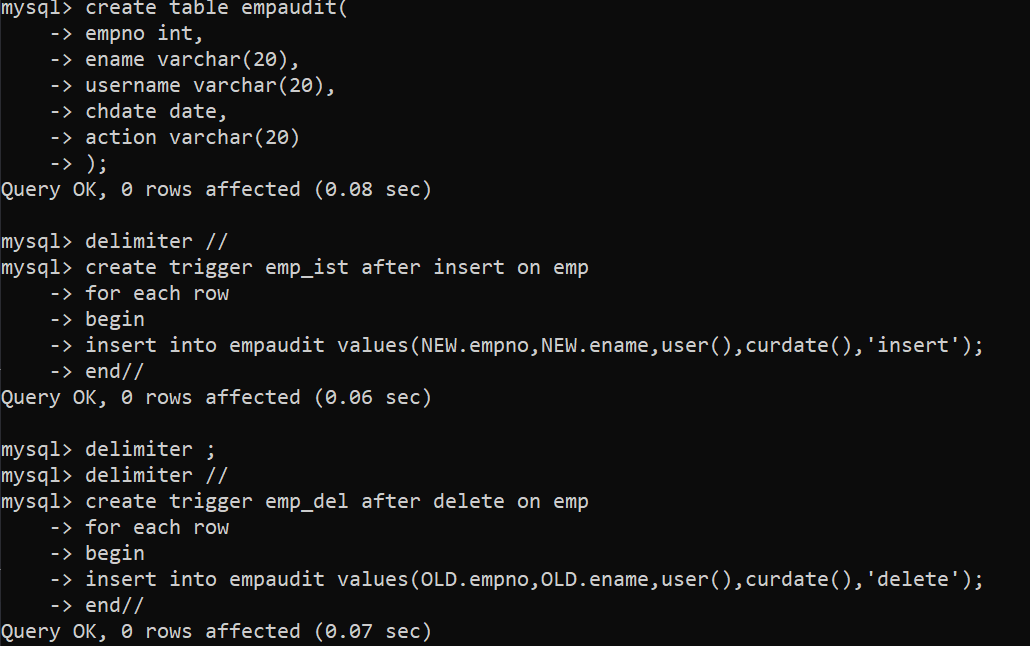
1) Write a tigger to store the old salary details in Emp \_Back (Emp \_Back has the same structure as emp table without any constraint) table

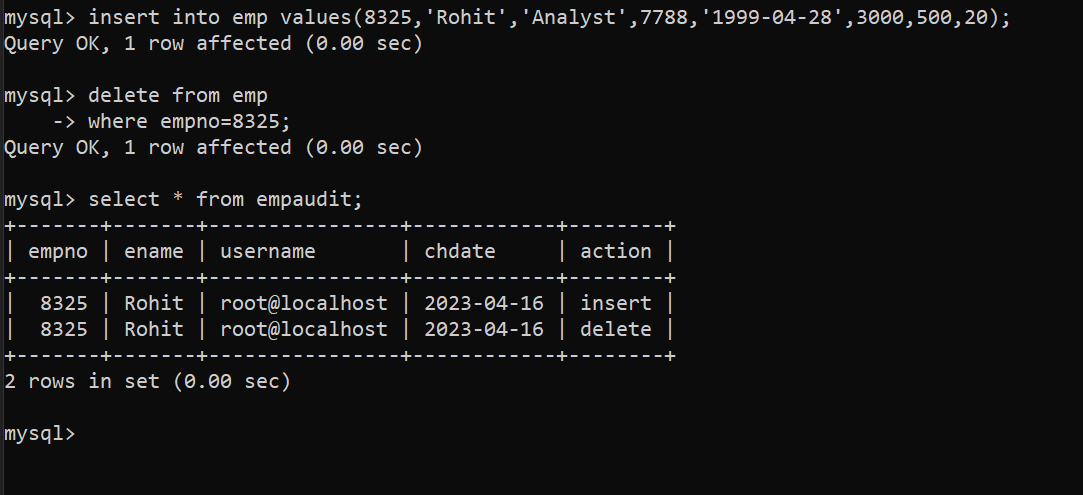




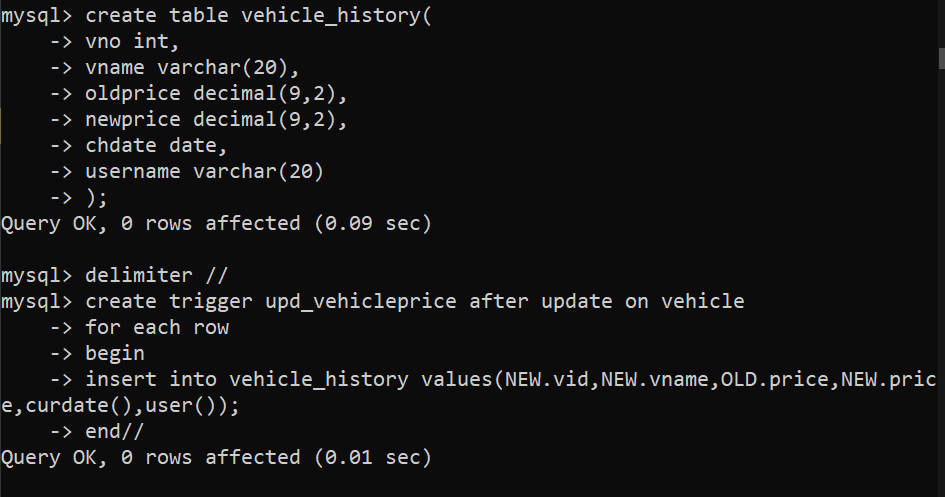


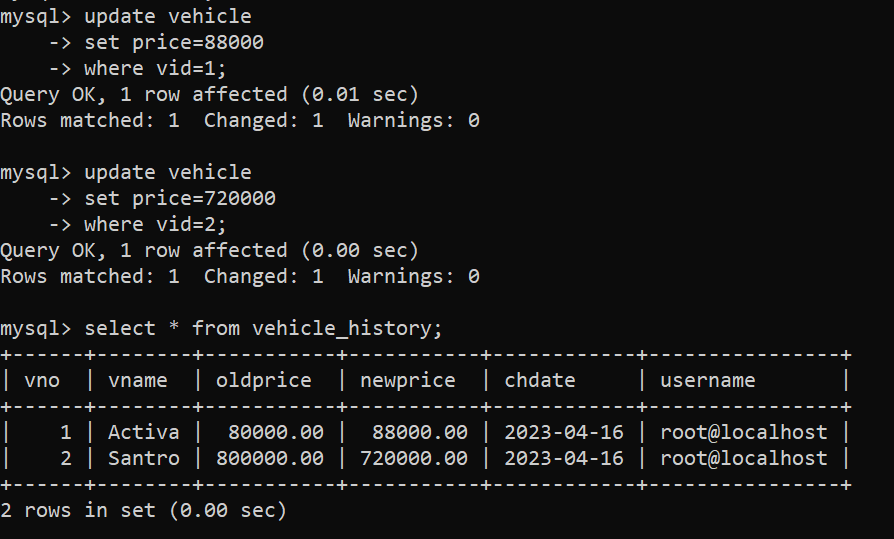
2) Write a trigger which add entry in audit table when user tries to insert or delete records in employee table store empno,name,username and date on which operation performed and which action is done insert or delete. in emp\_audit table. create table before writing trigge





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)



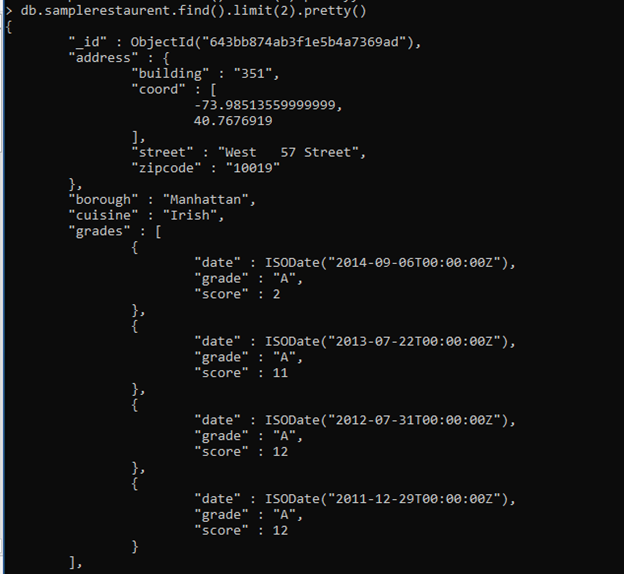


**MONGO DB**

**Assignment -1**

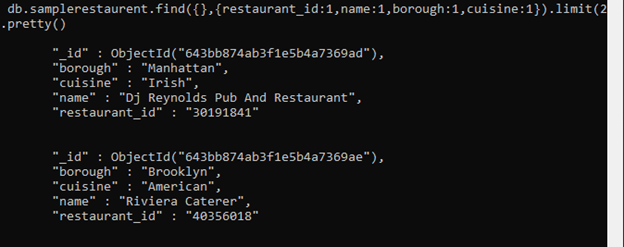
**1. Write a MongoDB query to display all the documents in the collection restaurants**

**=> db.samplerestaurent.find().limit(2).pretty()**

****

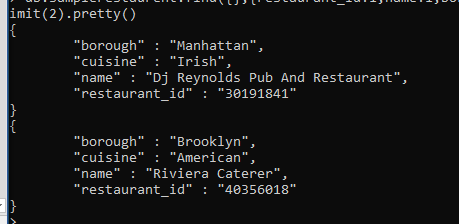
**2. Write a MongoDB query to display the fields restaurant\_id, name, borough and cuisine for all the documents in the collection restaurant.**

**=>db.samplerestaurent.find({},{restaurant\_id:1,name:1,borough:1,cuisine:1}).limit(2).pretty()**

****

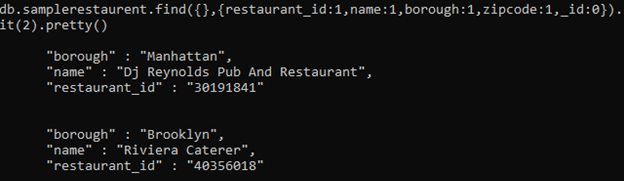
**3. Write a MongoDB query to display the fields restaurant\_id, name, borough and cuisine,but exclude the field \_id for all the documents in the collection restaurant.**

**\=>db.samplerestaurent.find({},{restaurant\_id:1,name:1,borough:1,cuisine:1,\_id:0}).limit(2).pretty()**

****

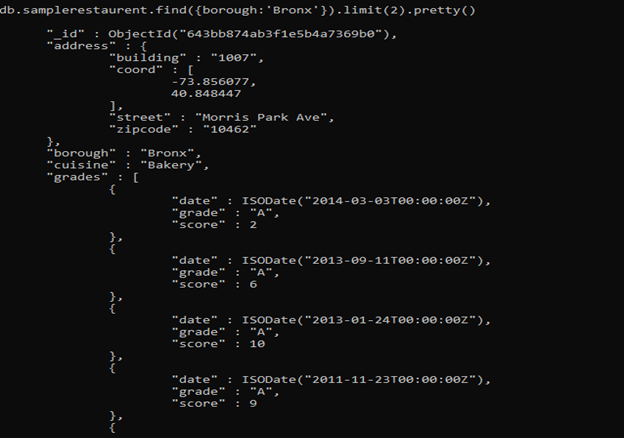
**4. Write a MongoDB query to display the fields restaurant\_id, name, borough and zip code, but exclude the field \_id for all the documents in the collection restaurant.**

**=>db.samplerestaurent.find({},{restaurant\_id:1,name:1,borough:1,zipcode:1,\_id:0}).limit(2).pretty()**

****

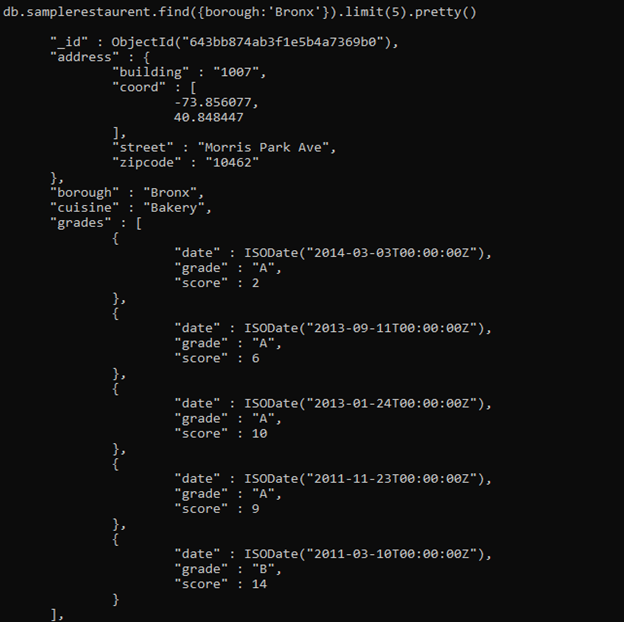
**5. Write a MongoDB query to display all the restaurant which is in the borough Bronx**

**=> db.samplerestaurent.find({borough:'Bronx'}).limit(2).pretty()**

****

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

=> db.samplerestaurent.find({borough:'Bronx'}).limit(5).pretty()



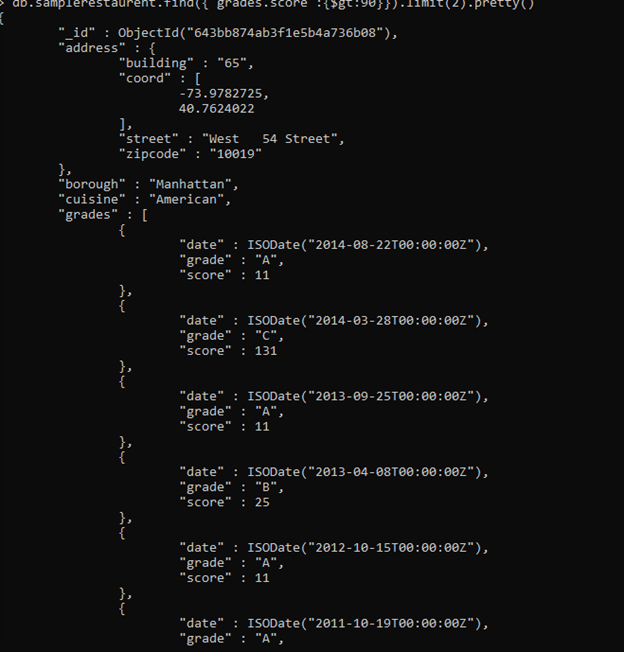
7.Write a MongoDB query to display the next 5 restaurants after skipping first 5 which are in the borough Bronx.

=> db.samplerestaurent.find({borough:'Bronx'}).skip(5).limit(5).pretty()



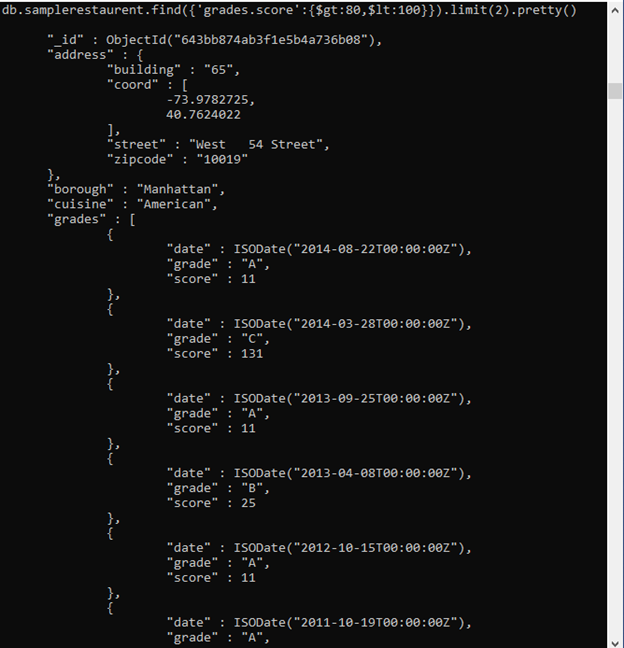
8. Write a MongoDB query to find the restaurants who achieved a score more than 90.

=> db.samplerestaurent.find({'grades.score':{$gt:90}}).limit(2).pretty()



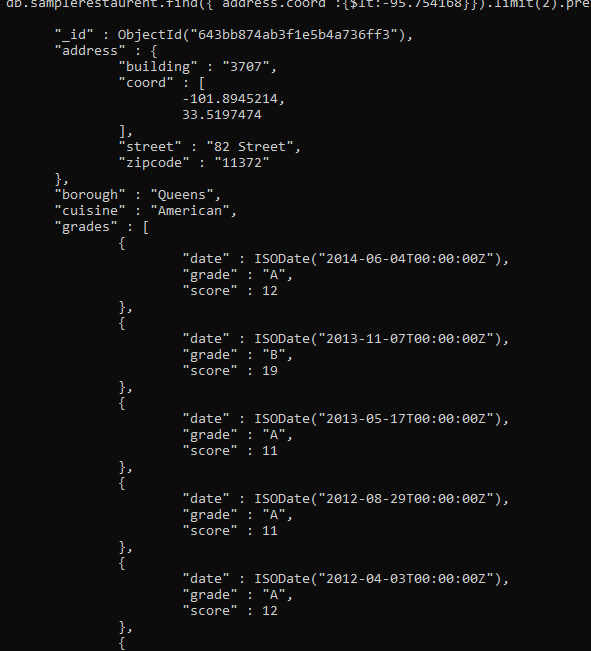
9. Write a MongoDB query to find the restaurants that achieved a score, more than 80 but less than 100.

=> db.samplerestaurent.find({'grades.score':{$gt:80,$lt:100}}).limit(2).pretty()



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

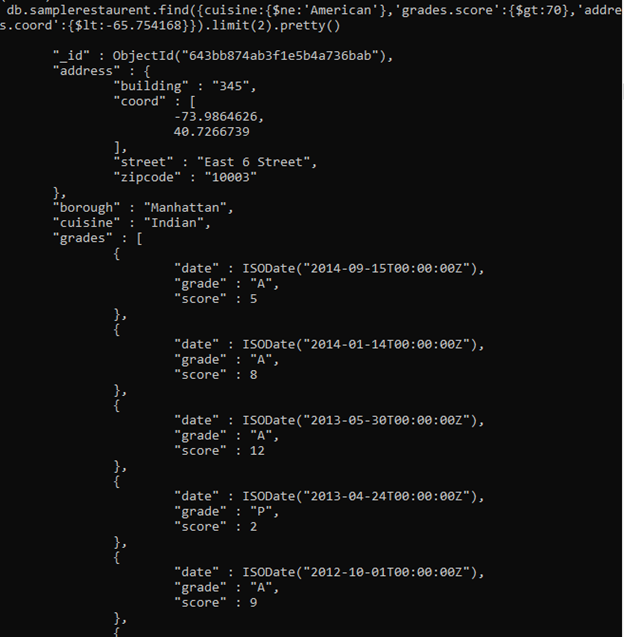
=> db.samplerestaurent.find({'address.coord':{$lt:-95.754168}}).limit(2).pretty()



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.

=>db.samplerestaurent.find({cuisine:{$ne:'American'},'grades.score':{$gt:70},'address.coord':{$lt:-65.754168}}).limit(2).pretty()



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.

=> db.samplerestaurent.find({cuisine:{$ne:'American'},'grades.score':{$gt:70},'address.coord.1':{$lt:65.754168}}).limit(2).pretty()



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.

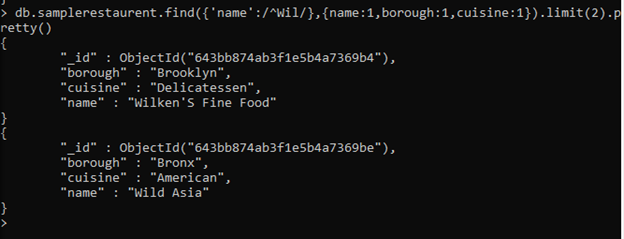
=> db.samplerestaurent.find({cuisine:{$ne:'American'},'grades.grade':'A',borough:{$ne:'Brooklyn'}}).limit(2).sort({cuisine:-1}).pretty()



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.

=> db.samplerestaurent.find({'name':/^Wil/},{name:1,borough:1,cuisine:1}).limit(2).pretty()



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.

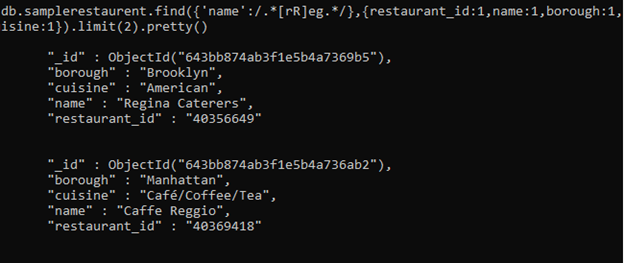
=>db.samplerestaurent.find({'name':/.\*ces$/},{restaurant\_id:1,name:1,borough:1,cuisine:1}).limit(2).pretty()



16. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those

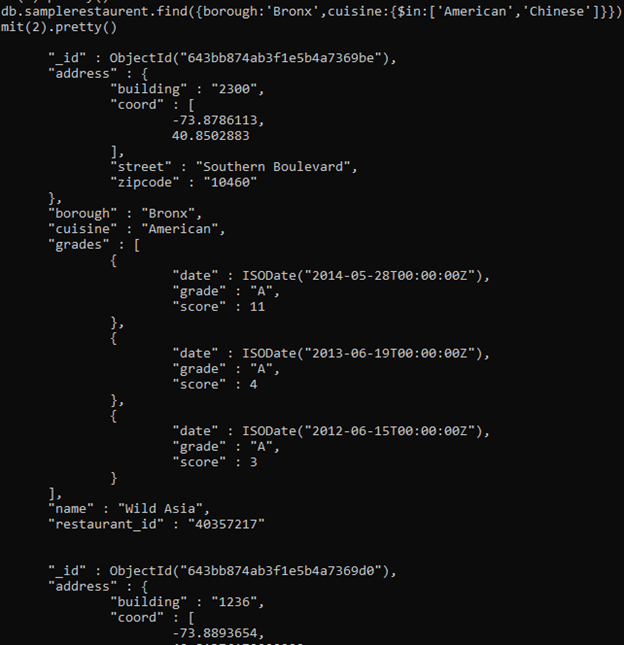
restaurants which contain 'Reg' as three letters somewhere in its name.

ð db.samplerestaurent.find({'name':/.\*[rR]eg.\*/},{restaurant\_id:1,name:1,borough:1,cuisine:1}).limit(2).pretty()



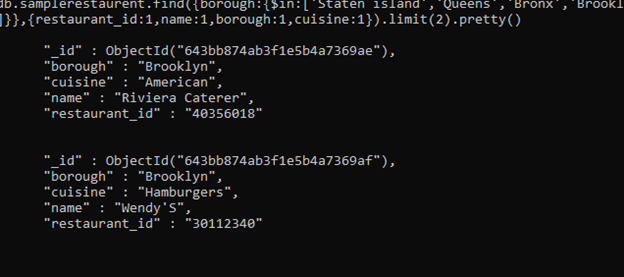
17. Write a MongoDB query to find the restaurants which belong to the borough Bronx and prepared either American or Chinese dish.

=>db.samplerestaurent.find({borough:'Bronx',cuisine:{$in:['American','Chinese']}}).limit(2).pretty()



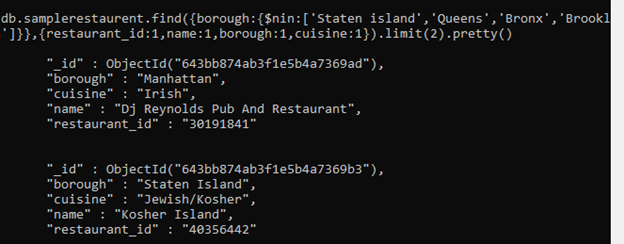
18. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which belong to the borough Staten Island or Queens or Bronxor Brooklyn.

=> db.samplerestaurent.find({borough:{$in:['Staten island','Queens','Bronx','Brooklyn']}},{restaurant\_id:1,name:1,borough:1,cuisine:1}).limit(2).pretty()



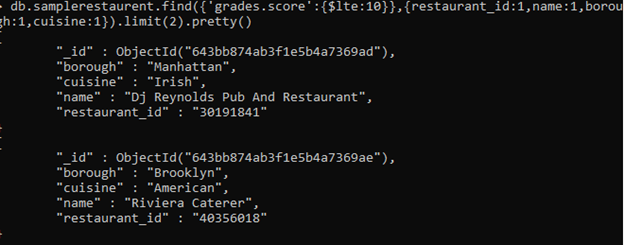
19. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which are not belonging to the borough Staten Island or Queens or Bronxor Brooklyn.

=> db.samplerestaurent.find({borough:{$nin:['Staten island','Queens','Bronx','Brooklyn']}},{restaurant\_id:1,name:1,borough:1,cuisine:1}).limit(2).pretty()



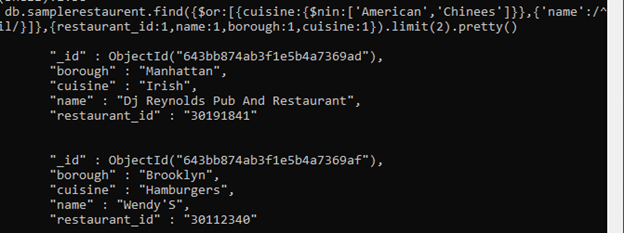
20. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which achieved a score which is not more than 10.

=>db.samplerestaurent.find({'grades.score':{$lte:10}},{restaurant\_id:1,name:1,borough:1,cuisine:1}).limit(2).pretty()



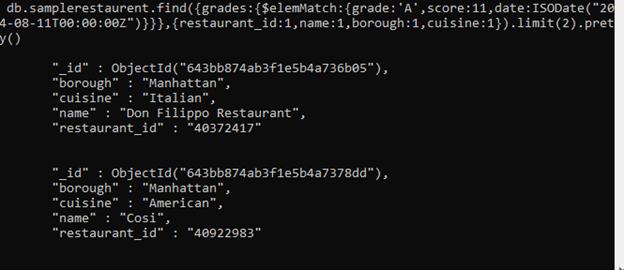
21. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which prepared dish except 'American' and 'Chinees' or restaurant's name begins with letter 'Wil'.

=>db.samplerestaurent.find({$or:[{cuisine:{$nin:['American','Chinees']}},{'name':/^Wil/}]},{restaurant\_id:1,name:1,borough:1,cuisine:1}).limit(2).pretty()

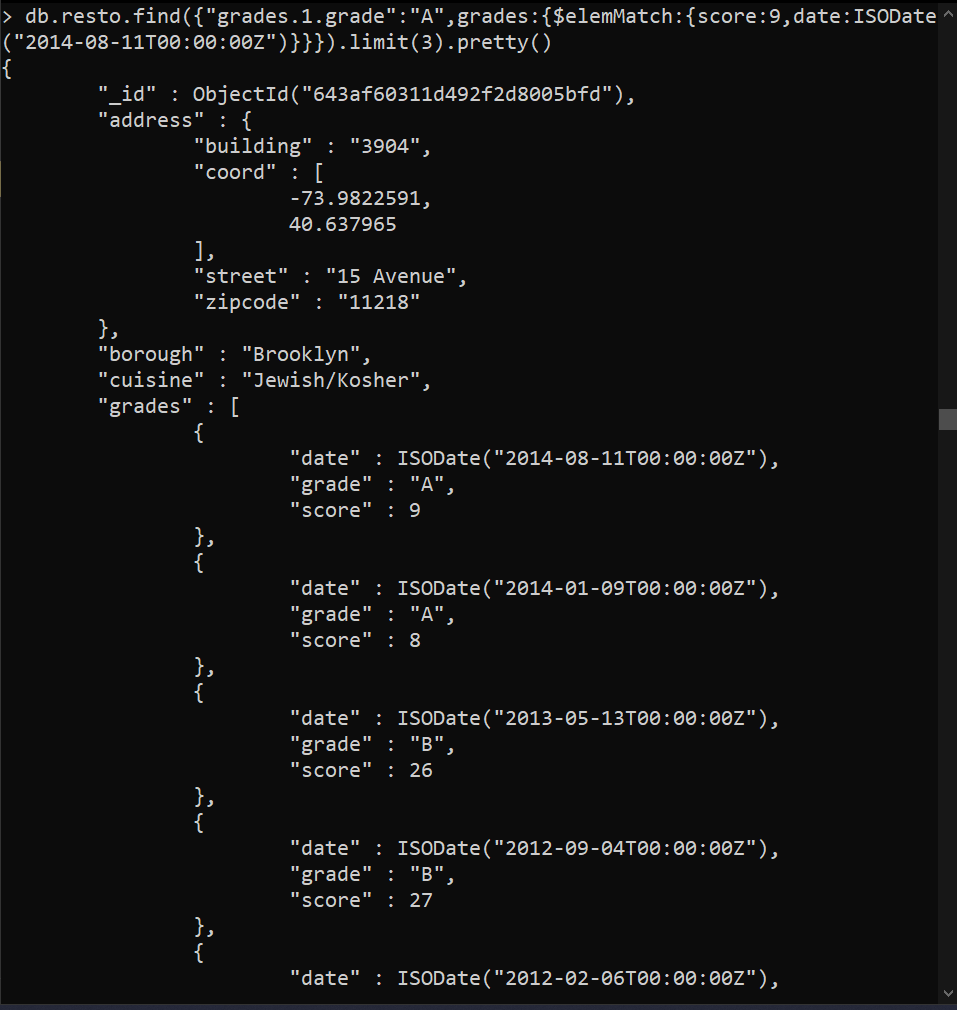


22. Write a MongoDB query to find the restaurant Id, name, and grades for those restaurants which achieved a grade of "A" and scored 11 on an ISODate "2014-08-11T00:00:00Z" among many of survey dates

=>db.samplerestaurent.find({grades:{$elemMatch:{grade:'A',score:11,date:ISODate("2014-08-11T00:00:00Z")}}},{restaurant\_id:1,name:1,borough:1,cuisine:1}).limit(2).pretty()

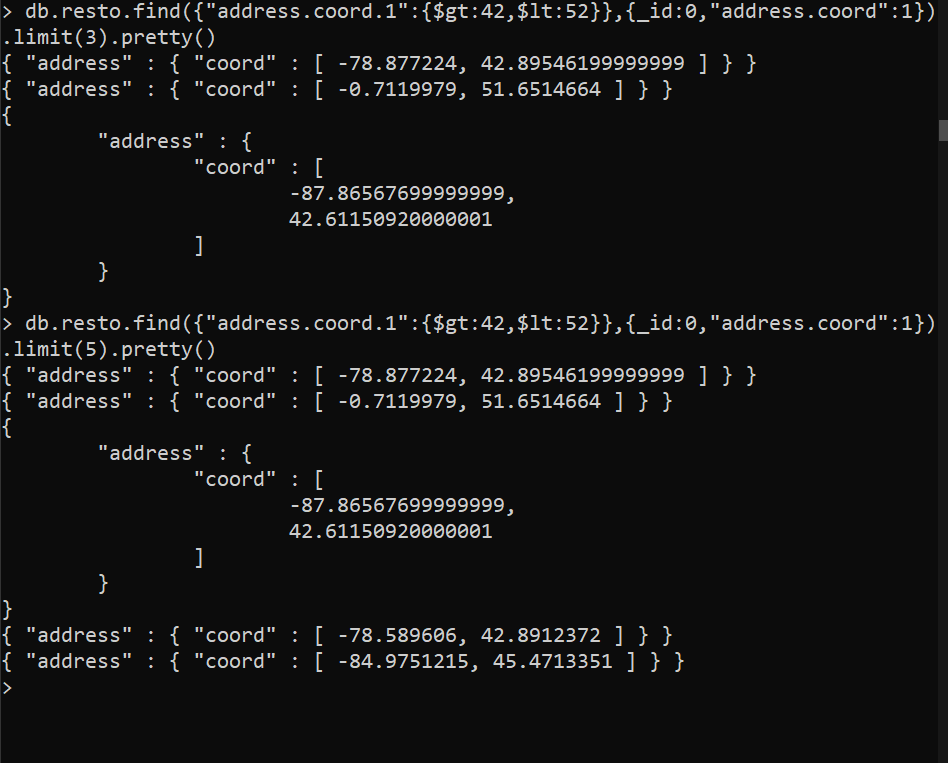


23. Write a MongoDB query to find the restaurant Id, name and grades for those restaurants where the 2nd element of grades array contains a grade of "A" and score 9 on an ISODate "2014-08-11T00:00:00Z".



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



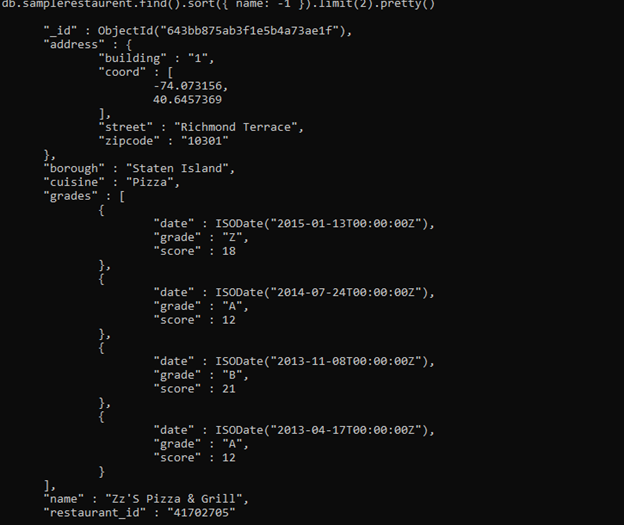
25. Write a MongoDB query to arrange the name of the restaurants in ascending order alongwith all the columns.

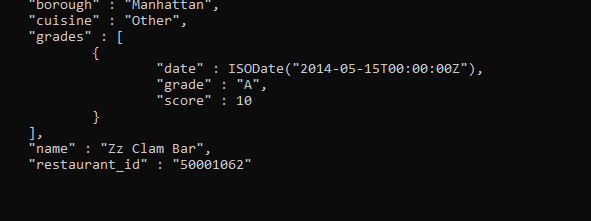
=> db.samplerestaurent.find().sort({ name: 1 }).limit(2).pretty()



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

=> db.samplerestaurent.find().sort({ name: -1 }).limit(2).pretty()





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.

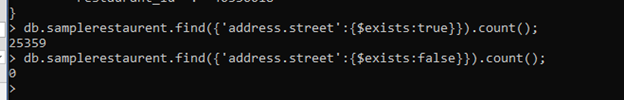
=> db.samplerestaurent.find().sort({name:1,borough:-1}).limit(2).pretty()



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

=> db.samplerestaurent.find({'address.street':{$exists:true}}).count();

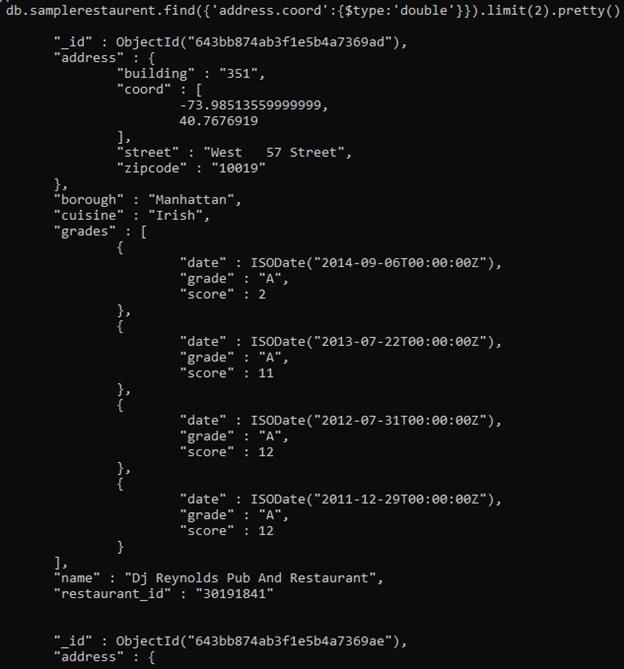
25359



29. Write a MongoDB query which will select all documents in the restaurants collection

where the coord field value is Double.

=> db.samplerestaurent.find({'address.coord':{$type:'double'}}).limit(2).pretty()



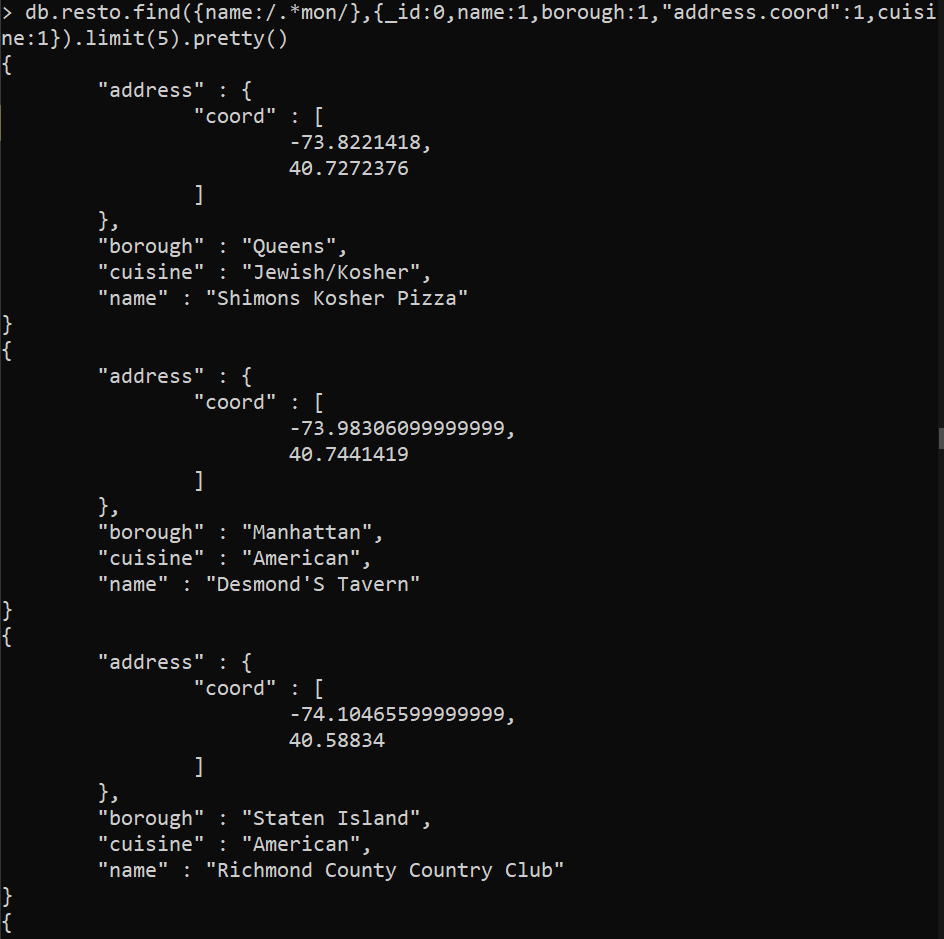
30. Write a MongoDB query which will select the restaurant Id, name and grades for those

restaurants which returns 0 as a remainder after dividing the score by 7.

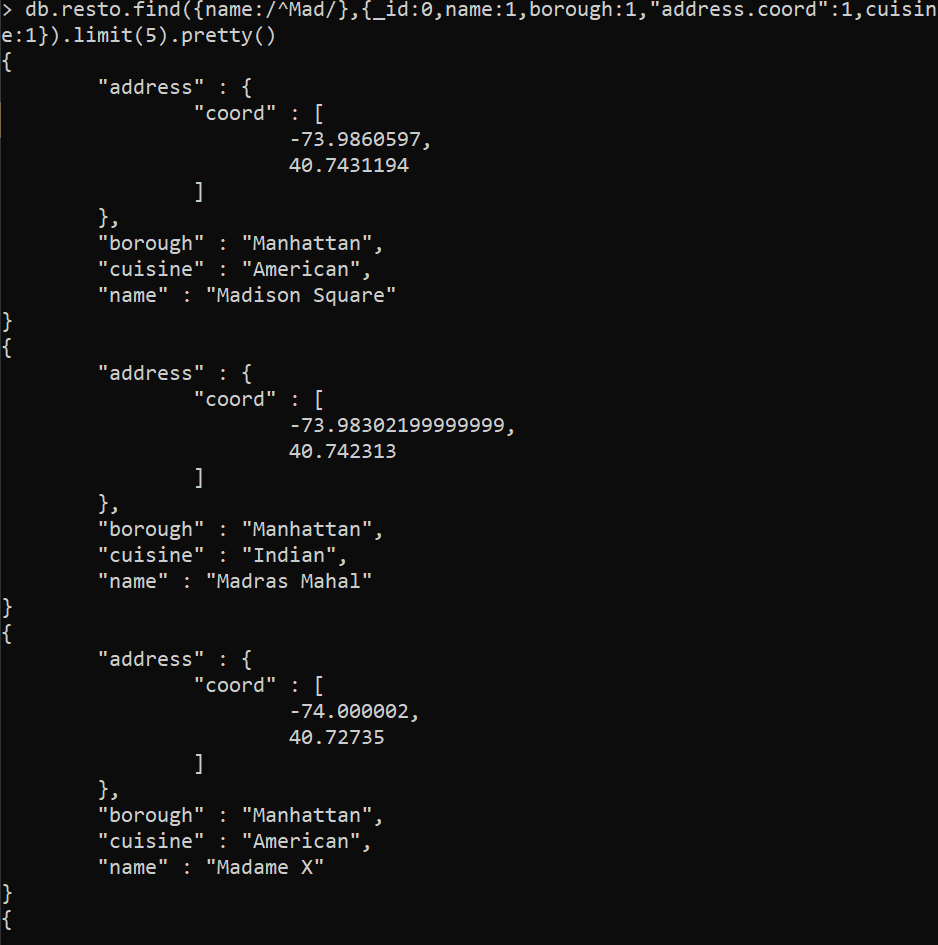
ð db.samplerestaurent.find({'grades.score':{$mod:[7,0]}},{restaurant\_id:1,name:1,'grades.score':1}).limit(2).pretty()



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.



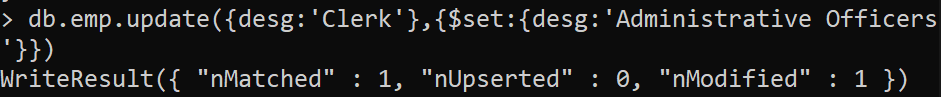
32. Write a MongoDB query to find the restaurant name, borough, longitude and latitude and cuisine for those restaurants which contain 'Mad' as first three letters of its name.



**MONGO DB**

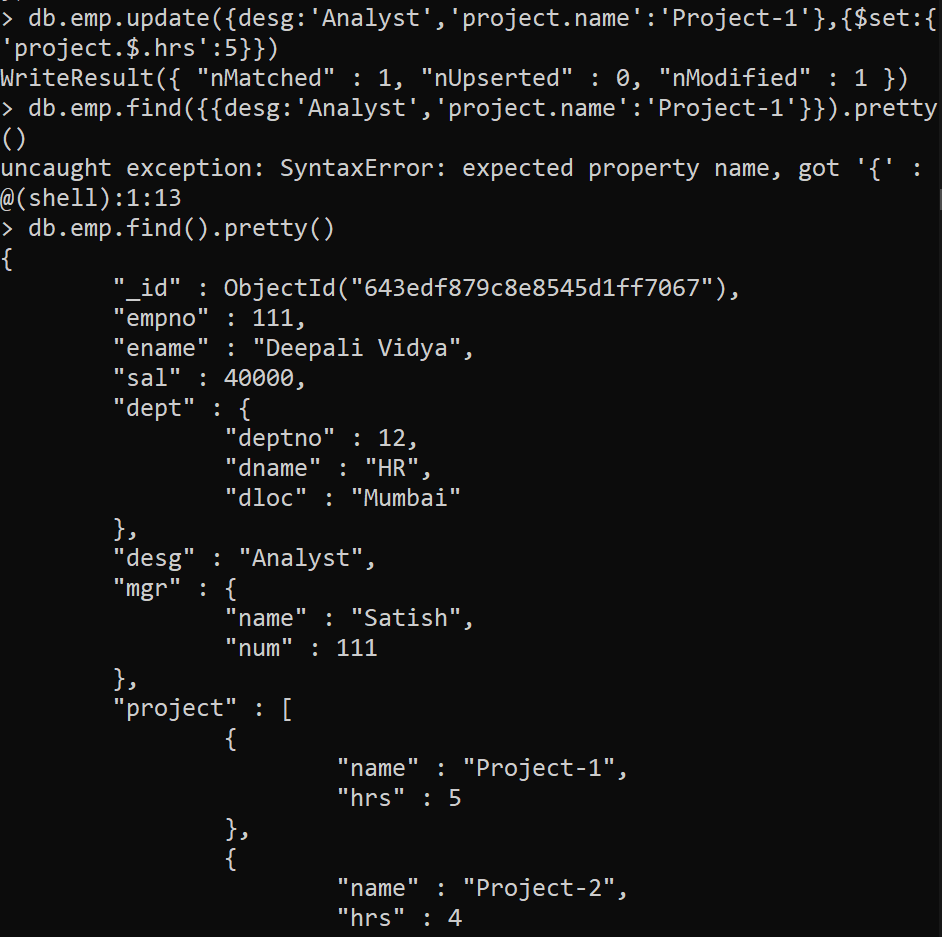
**ASSIGNMENT-2**

1) All Employee’s with the desg as ‘CLERK’ are now called as (AO) Administrative Officers. Update the Employee collection for this





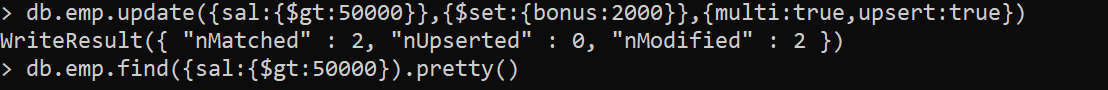
2) Change the number of hours for project-1 to 5 for all employees with designation analyst.

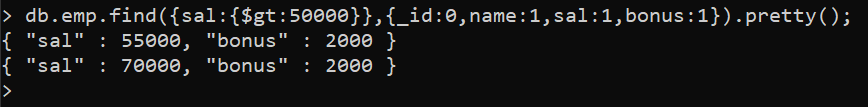


3) Add 2 projects project-3 and project-4 for employee whose name starts with ”Deep” with 2 hrs

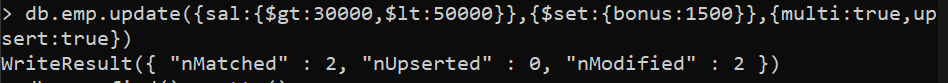


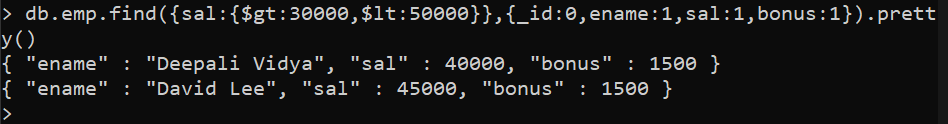
4. Add bonus rs 2000 for all employees with salary > 50000





5. Add bonus rs 1500 if salary 30000

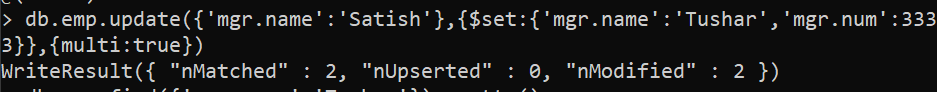


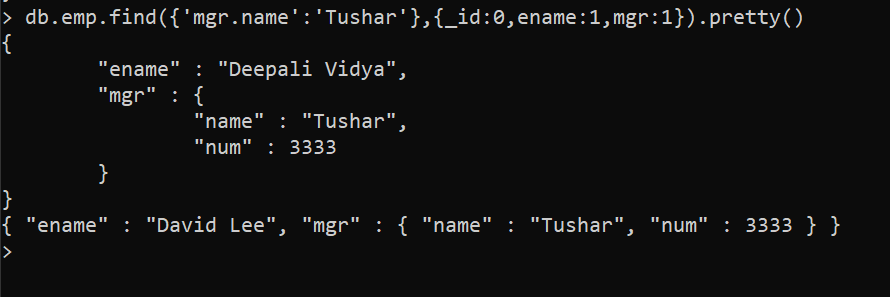


6. increment bounus by 1000 for all employees if salary <=30000

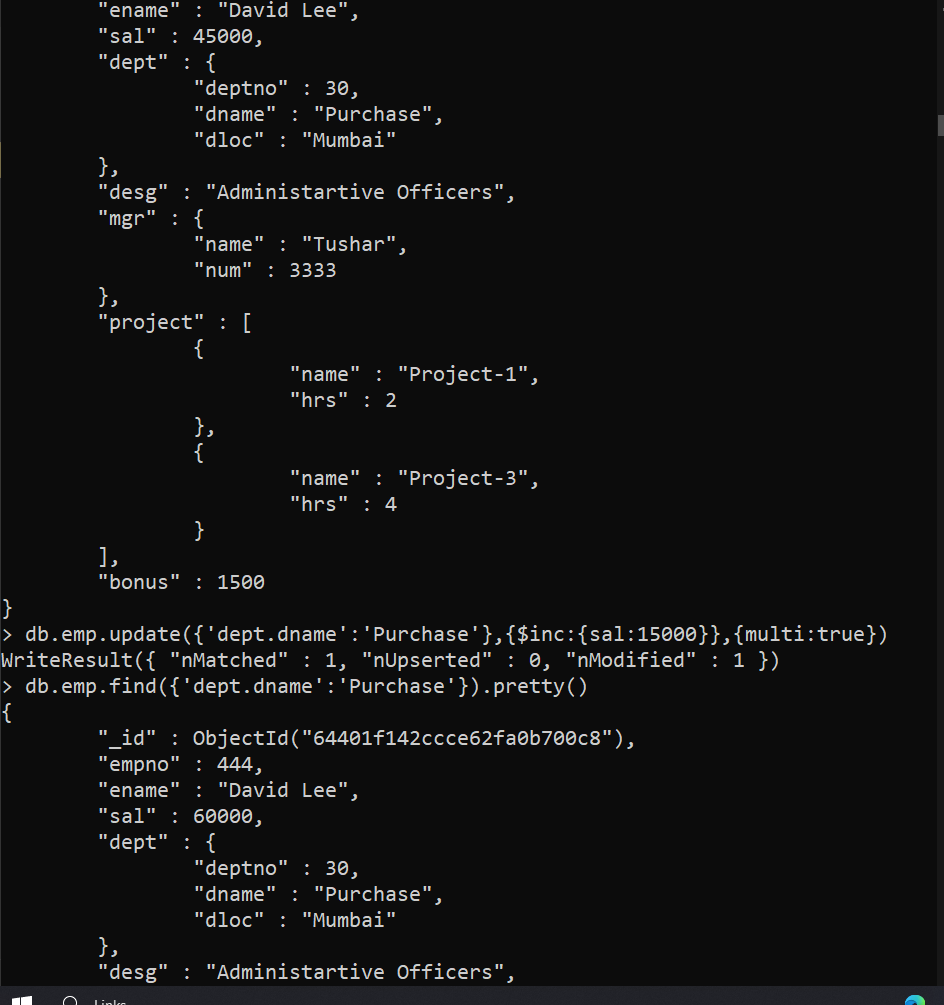


7. Change manager name to Tushar for all employees whose manager is currently “satish” And manager number to 3333

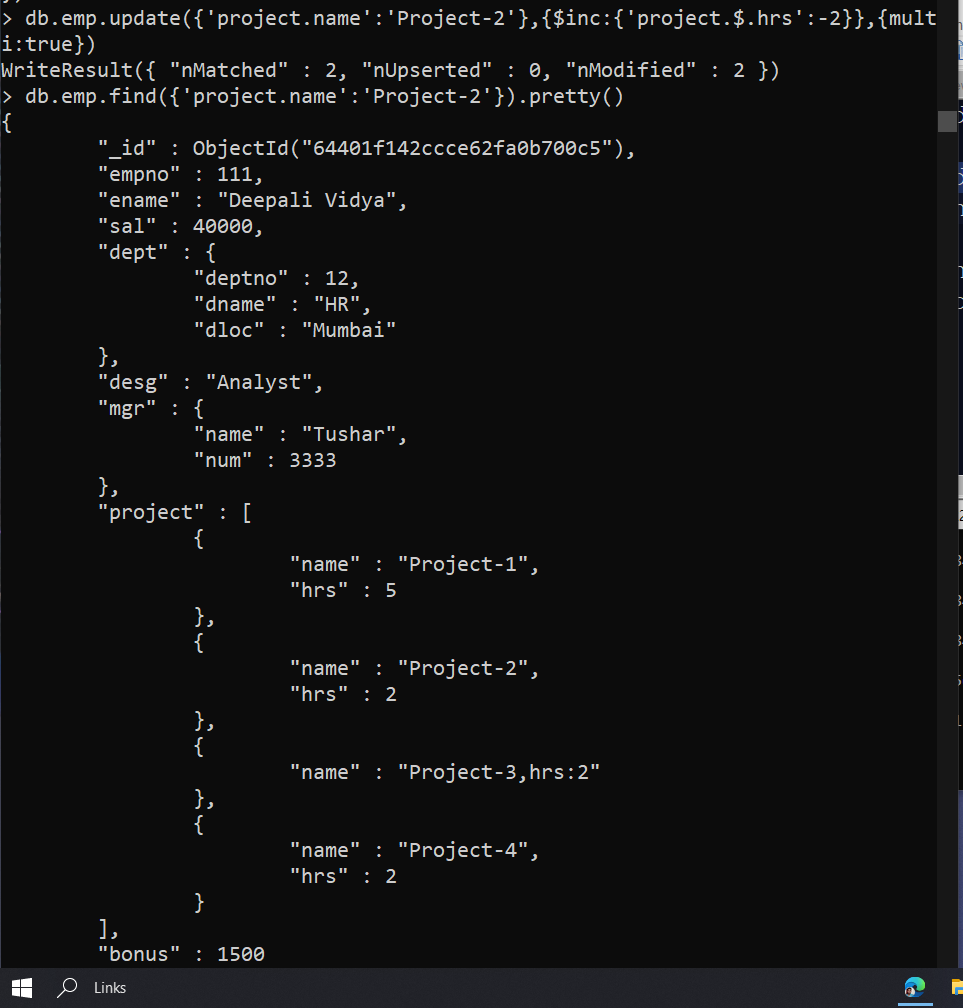




8. Increase salary of all employees from “purchase department” by 15000



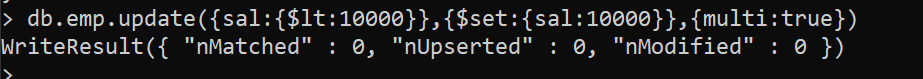
9. Decrease number of hrs by 2 for all employees who are working on project-2



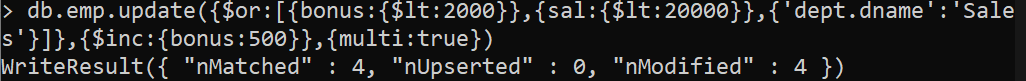
10. Delete project-2 from all employee document if they are working on the project for 4 hrs.



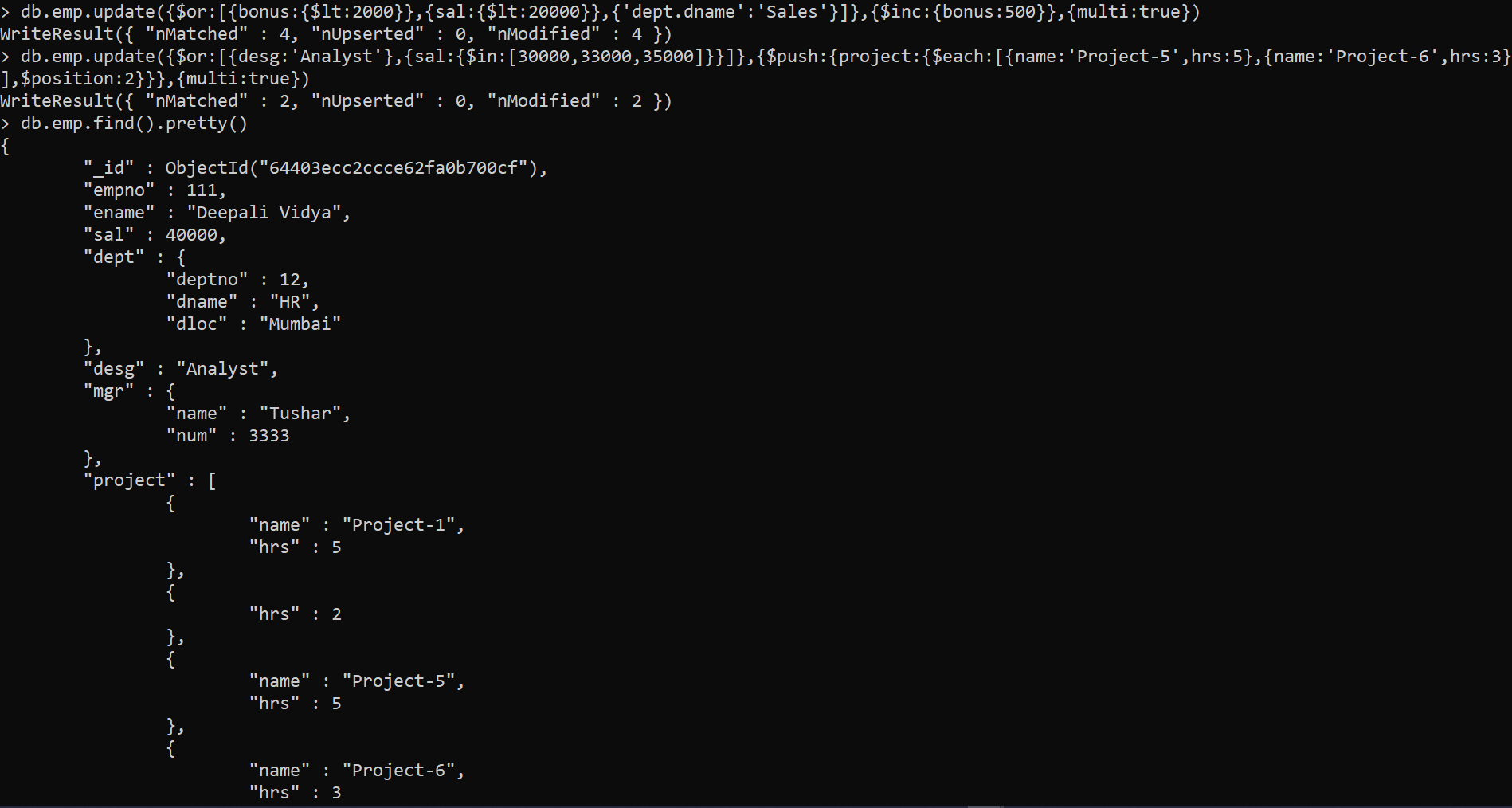
11. Change the salary of employees to 10000 only if their salary is < 10000



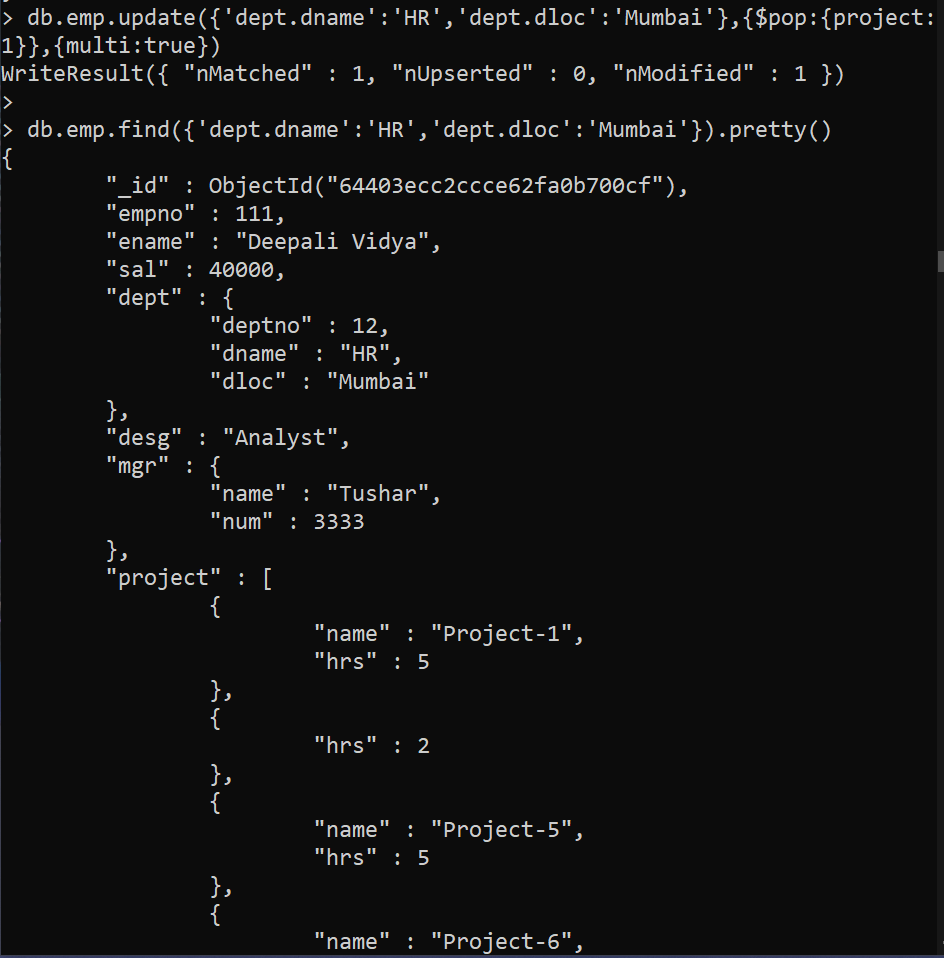
12. Increase bonus of all employees by 500 if the bonus is < 20000 or if employee belong to sales department



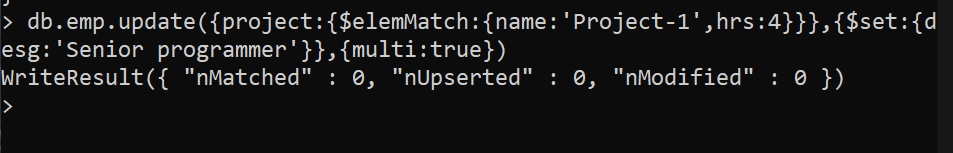
13. Add 2 new project at position 2 for all employees with designation analyst or salary is equal to either 30000 or 33000 or 35000



14. Delete last project of all employees with department name is “HR” and if the location is Mumbai



15. Change designation of all employees to senior programmer if they are working on name:”Project-1” for 4 hrs



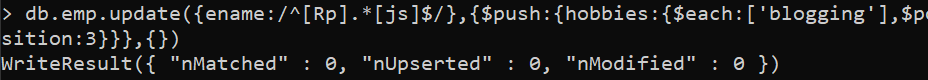
16. Add list of hobbies in all employees document whose manager is Rajan or Revati



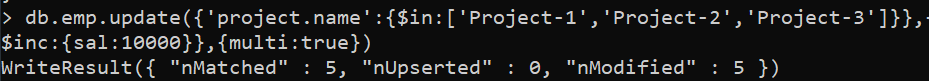
17. Add list of skillset in all employee documents who are working on project-4 for 3 hrs or on project-3 for 4 hrs



18. Add a new hobby as blogging at 3 position in hobbies array for all employess whose name starts with R or p and ends with j or s

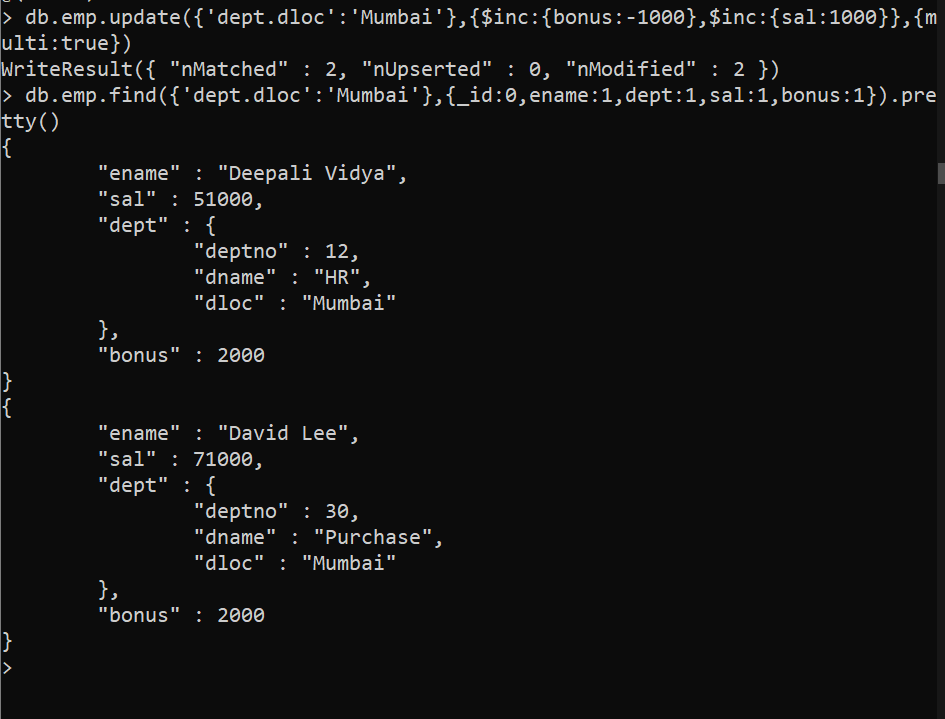


19. Increase salary by 10000 for all employees who are working on project-2 or project-3 or project-1

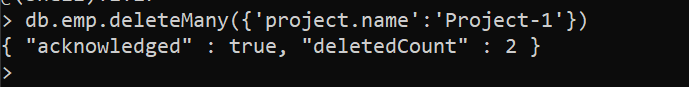


Ques)

Decrease bonus by 1000 rs And increase salary by 1000rs for all employees whose department location is Mumbai



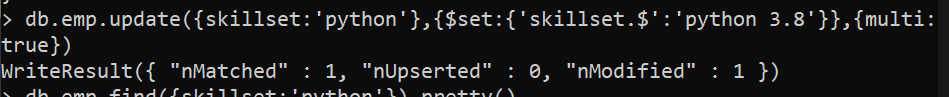
20. Remove all employees working on project-1

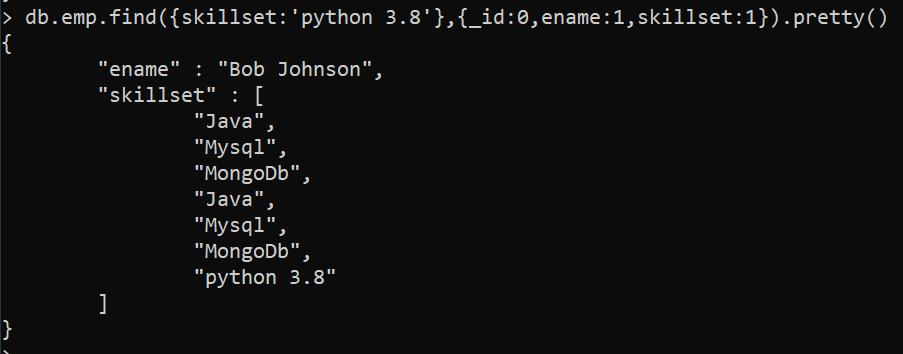


21. Replace document of employee with name “Deepak to some new document



22. Change skill python to python 3.8 for all employees if python is there in the skillset





23. Add 2 skills MongoDb and Perl at the end of skillset array for all employees who are working at Pune location .



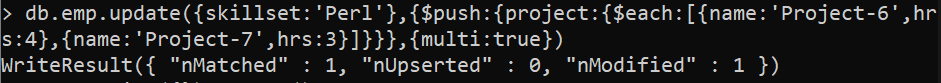
24. Delete first hobby from hobby array for all employees who are working on project-1 or project-2



25. Delete last hobby from hobbies array for all employees who are working on project which is at 2 nd position in projects array for 4 hrs



26. Add 2 new projects at the end of array for all employees whose skillset contains Perl or python





27. Change hrs to 6 for project-1 for all employees if they working on the project-1 for < 6 hrs. otherwise keep the existing value

