**DAY-3 ASSIGNMENT**

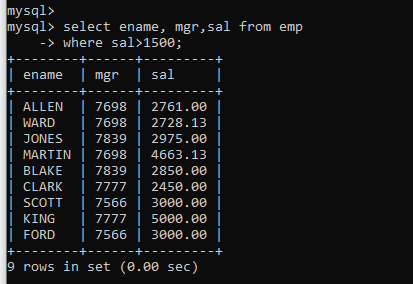
**PRATIKSHA TAMBE(243072)**

**Write SQL statement for the following**

1. To find all managers with salary >1500

**select ename, mgr,sal from emp**

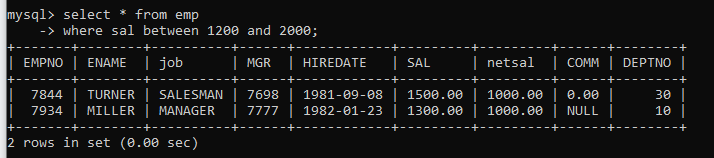
**-> where sal>1500;**

****

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

**select \* from emp**

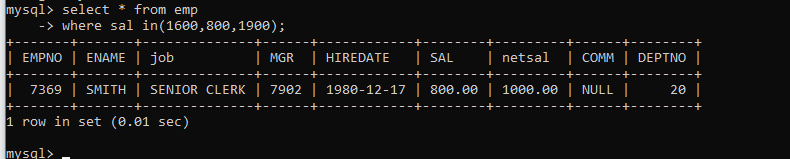
**-> where sal between 1200 and 2000;**

****

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

**select \* from emp**

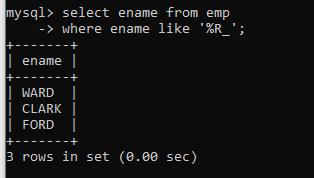
**-> where sal in(1600,800,1900);**



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

**select ename from emp**

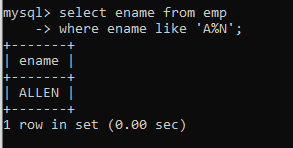
**-> where ename like '%R\_';**



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

**select ename from emp**

**-> where ename like 'A%N';**

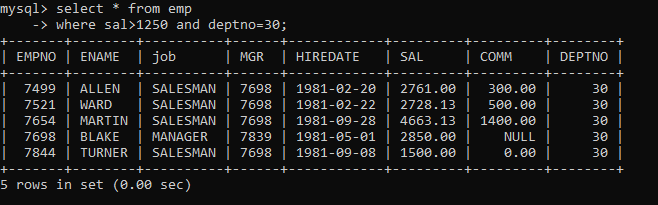


**Q2. Solve following**

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

**select \* from emp**

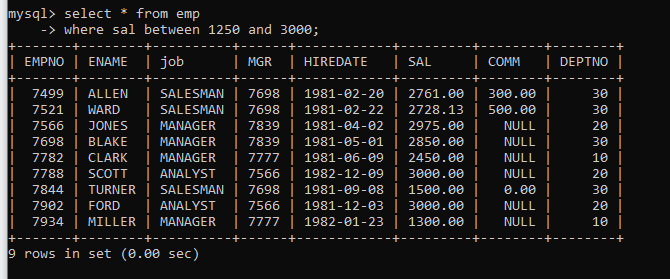
**where sal>1250 and deptno=30;**

****

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

**select \* from emp**

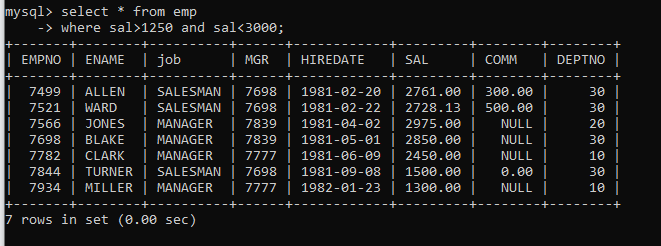
**where sal between 1250 and 3000;**

****

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

**select \* from emp**

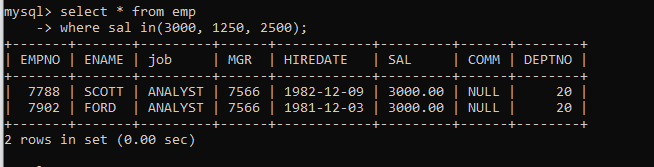
**where sal>1250 and sal<3000;**

****

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

**select \* from emp**

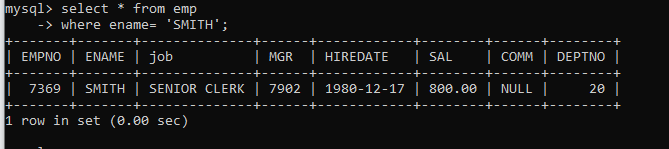
**where sal in(3000, 1250, 2500);**

****

5. list all employee with name=SMITH

**select \* from emp**

**where ename= ‘SMITH’;**

****

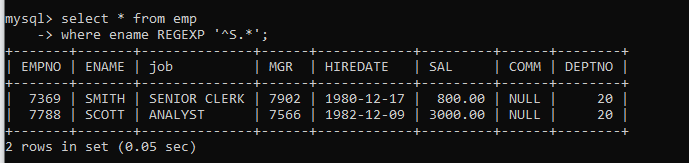
6. list all employees with name starting with S

**select \* from emp**

**where ename like ‘S%’;**

**OR  
select \* from emp**

**where ename REGEXP ‘^S.\*’;**

****

7. list all employees with name ending with S

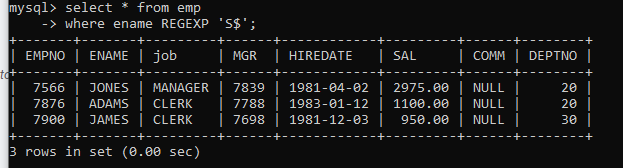
**select \* from emp**

**where ename like ‘S%’;**

**OR**

**select \* from emp**

**where ename REGEXP ‘S$’;**

****

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

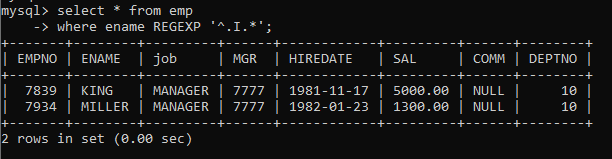
**select \* from emp**

**where ename like ‘\_I%’;**

**OR**

**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

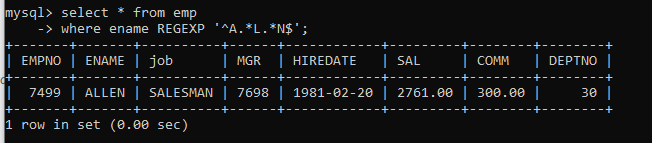
**select \* from emp**

**where ename like ‘A%L%N’;**

**OR**

**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

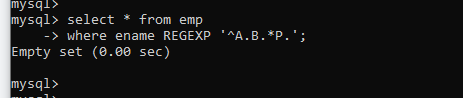
**select \* from emp**

**where ename like ‘A%\_B%P\_’;**

**OR**

**select \* from emp**

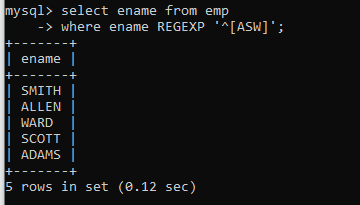
**where ename REGEXP ‘^A.B.\*P.’;**

****

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

**select ename from emp**

**-> where ename REGEXP '^[ASW]';**

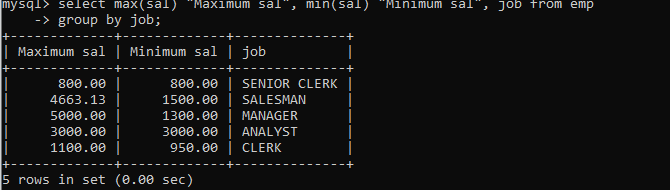
****

**practice Aggregate functions**

12. find max sal and min sal for each job

**select max(sal) "Maximum sal", min(sal) "Minimum sal", job from emp**

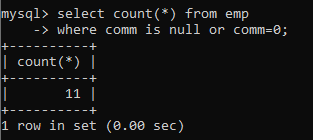
**-> group by job;**



13. find how many employess have not received commission

**select count(\*) from emp**

**-> where comm is null or comm=0;**

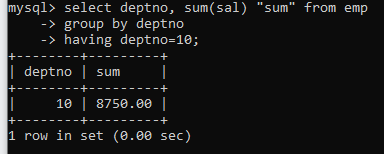
****

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

**select deptno, sum(sal) "sum" from emp**

**-> group by deptno**

**-> having deptno=10;**

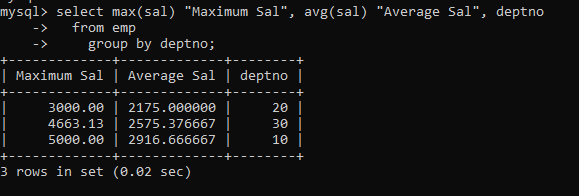
****

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

**select max(sal) "Maximum Sal", avg(sal) "Average Sal", deptno**

**-> from emp**

**-> group by deptno;**

****

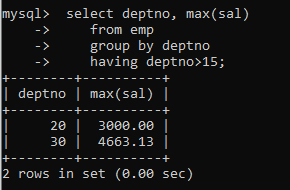
16. find max salary for every department if deptno is > 15 and arrange data in deptno order.

**select deptno, max(sal)**

**-> from emp**

**-> group by deptno**

**-> having deptno>15;**

****

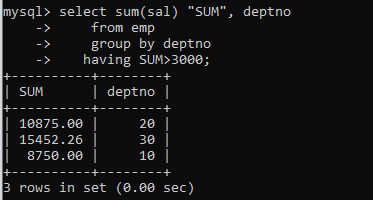
17. find sum salary for every department if sum is > 3000

**select sum(sal) "SUM", deptno**

**-> from emp**

**-> group by deptno**

**-> having SUM>3000;**

****

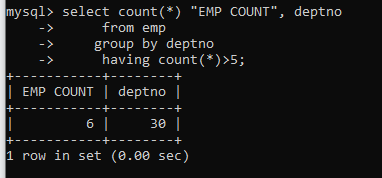
18. list all department which has minimum 5 employees

**select count(\*) "EMP COUNT", deptno**

**-> from emp**

**-> group by deptno**

**-> having count(\*)>5;**

****

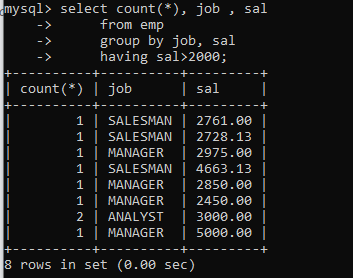
19. count how many employees earn salary more than 2000 in each job

**select count(\*), job , sal**

**-> from emp**

**-> group by job, sal**

**-> having sal>2000;**

****

20. list all enames and jobs in small case letter

**select lower(ename), lower(job)**

**-> from emp;**

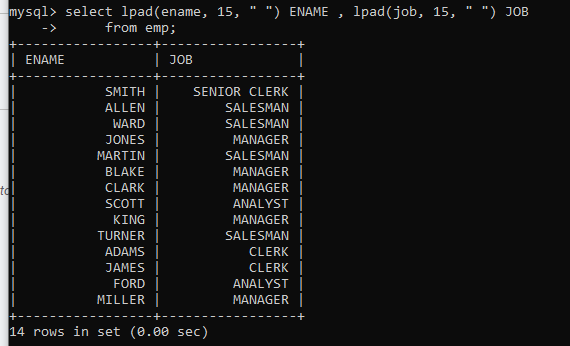
****

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, " ") ENAME , lpad(job, 15, " ") JOB**

**-> from emp;**

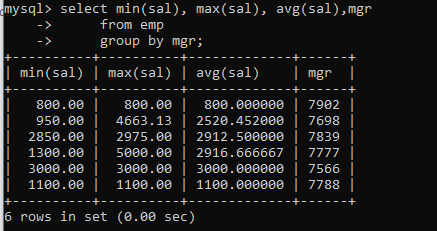
****

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

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

**-> from emp**

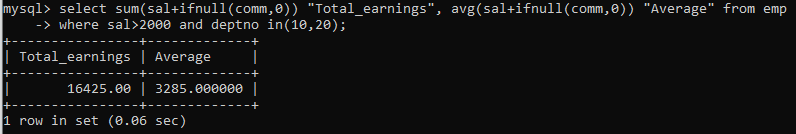
**-> group by mgr;**

****

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

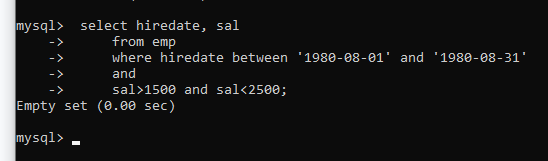
**select hiredate, sal**

**-> from emp**

**-> where hiredate between '1980-08-01' and '1980-08-31'**

**-> and**

**-> sal>1500 and sal<2500;**

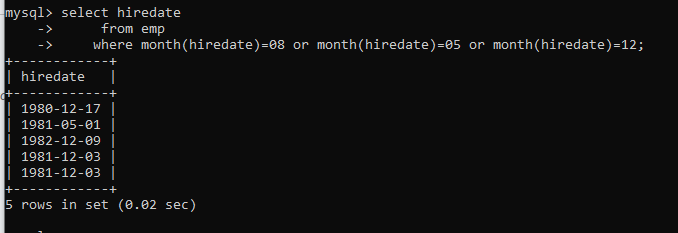
****

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

**select hiredate**

**-> from emp**

**-> where month(hiredate)=08 or month(hiredate)=05 or month(hiredate)=12;**

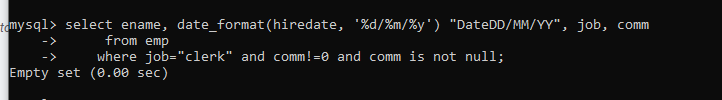
****

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') "DateDD/MM/YY", job, comm**

**-> from emp**

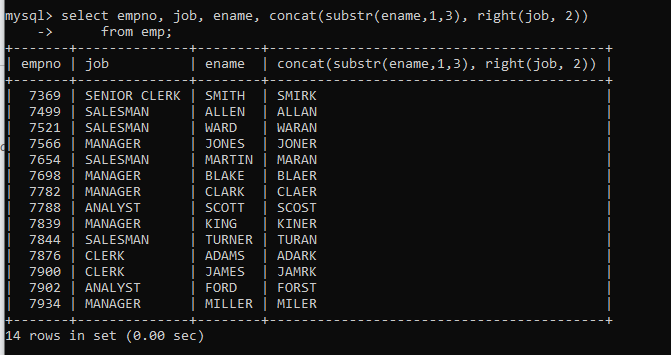
**-> where job="clerk" and comm!=0 and comm is not null;**

****

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 empno, job, ename, concat(substr(ename,1,3), right(job, 2))**

**-> from emp;**

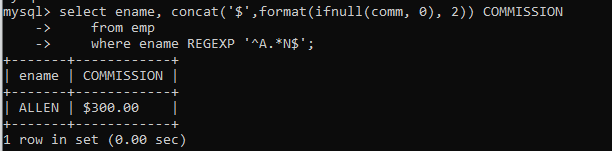
****

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

**select ename, concat('$',format(ifnull(comm, 0), 2)) COMMISSION**

**-> from emp**

**-> where ename REGEXP '^A.\*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**

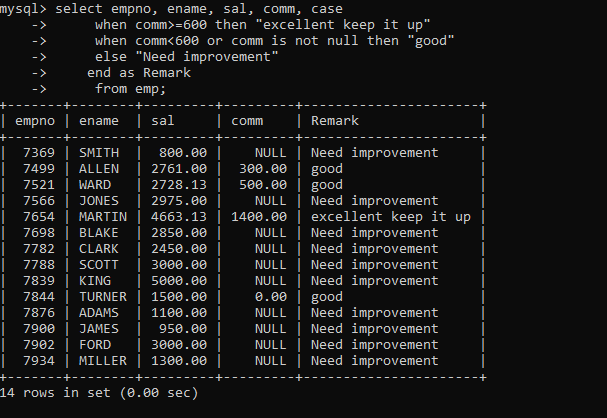
**-> when comm>=600 then "excellent keep it up"**

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

**-> else "Need improvement"**

**-> end as 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,deptno, ename,**

**-> case**

**-> when deptno=10 then "Hr"**

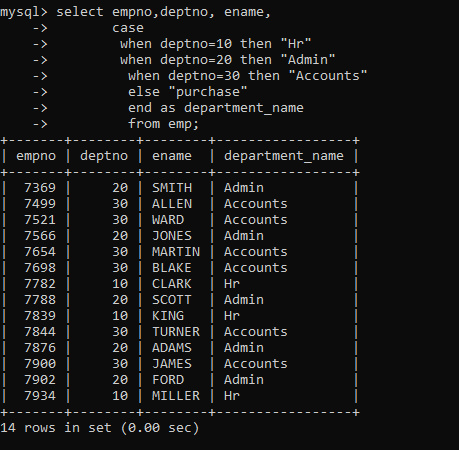
**-> when deptno=20 then "Admin"**

**-> when deptno=30 then "Accounts"**

**-> else "purchase"**

**-> end as department\_name**

**-> from emp;**

****

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

31. Practice creating following tables

MySQL syntax:

create table mydept\_DBDA

(

deptid int primary key,

dname varchar(20) not null unique,

dloc varchar(20)

)

Oracle syntax:

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');

MySql syntax:

create table myemployee

(

empno int primary key,

fname varchar(15) not null,

mname varchar(15),

lname varchar(15) not null,

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

doj date,

passportnum varchar(15) unique,

deptno int,

constraint fk\_deptno foreign key(deptno) references mydept\_DBDA(deptid) on

delete set null

on update cascade

)

Oracle syntax:

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

)

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;

)

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

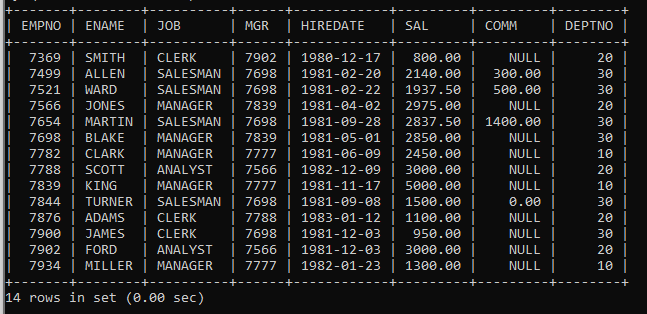
deptno --->> dept(deptno)

add new column in emp table netsal with constraint default 1000

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

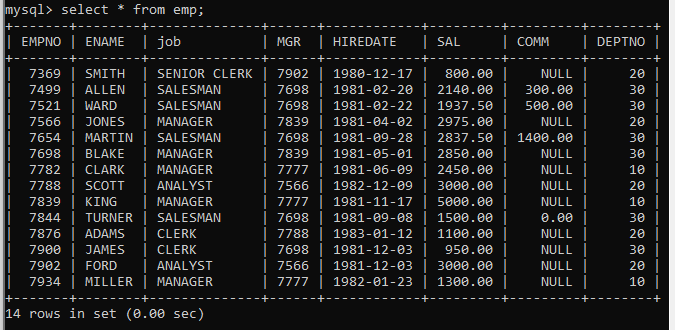
manager and mgr to 7777 for all employees in deptno 10.

**update emp  
set sal=sal+0.15\*sal+comm  
where ifnull(comm,0);  
  
update emp  
set job ='MANAGER',  
mgr=7777  
where deptno=10;**

****

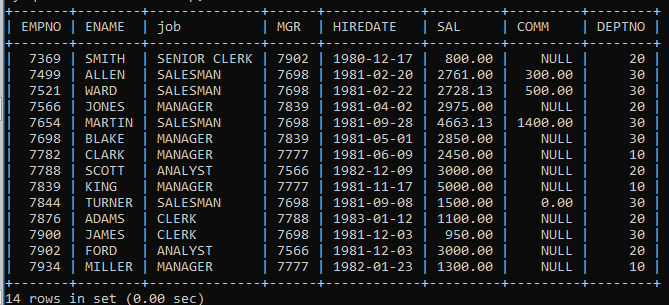
36. change job of smith to senior clerk

**alter table emp  
modify job varchar(20);  
for(ERROR 1406 (22001): Data too long for column 'JOB' at row 1) This Error  
  
  
update emp  
set job='SENIOR CLERK'  
where ename='SMITH';**

****

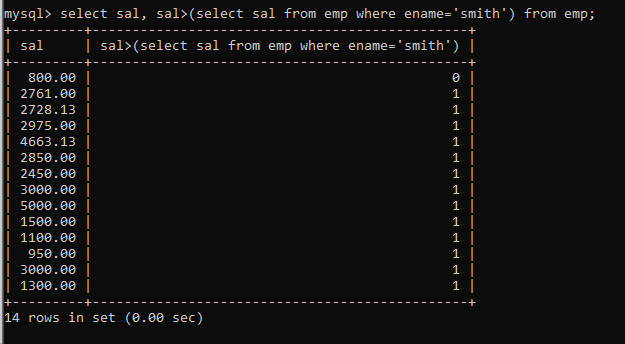
37. increase salary of all employees by 15% if they are earning some commission

**update emp  
set sal=sal+0.15\*sal+comm  
where comm is not null and comm!=0;**

****

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

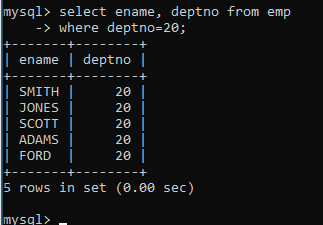
**select sal, ename from emp  
where ename!='smith' and sal>800;  
  
OR  
  
select sal, sal>(select sal from emp where ename='smith') from emp;**

****

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

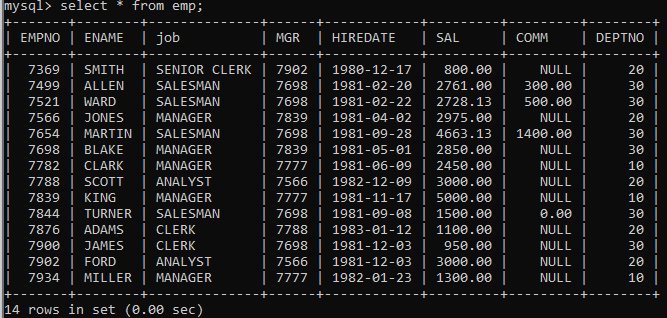
**select ename, deptno from emp  
where deptno=20;**

**OR  
select ename, deptno 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 sal, ename from emp  
where ename=(select ename from emp where ename='rajan' and ename='revati')  
and sal>(select ename from emp where ename='rajan' and ename='revati');**

****

41. delete all employees working in alan's department

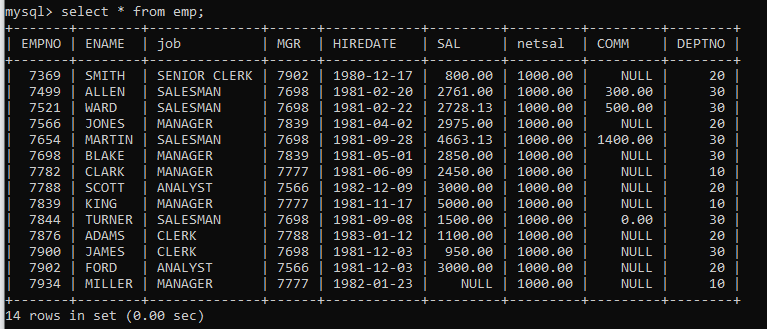
42. change salary of Alan to the salary of Miller.

**update emp**

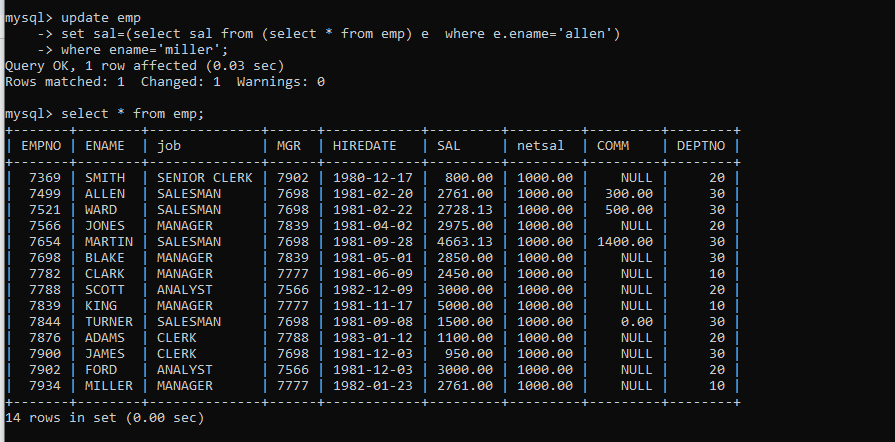
**-> set sal=(select sal from (select \* from emp) e where e.ename='allen')**

**-> where ename='miller';**

**before update table**

****

**after update table**

****

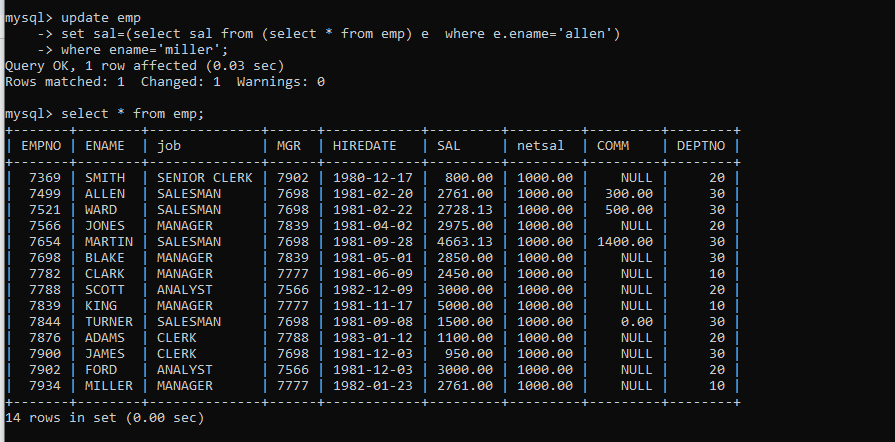
43. change salary of all employees who working in Ward's department to the salary of Miller.

**update emp**

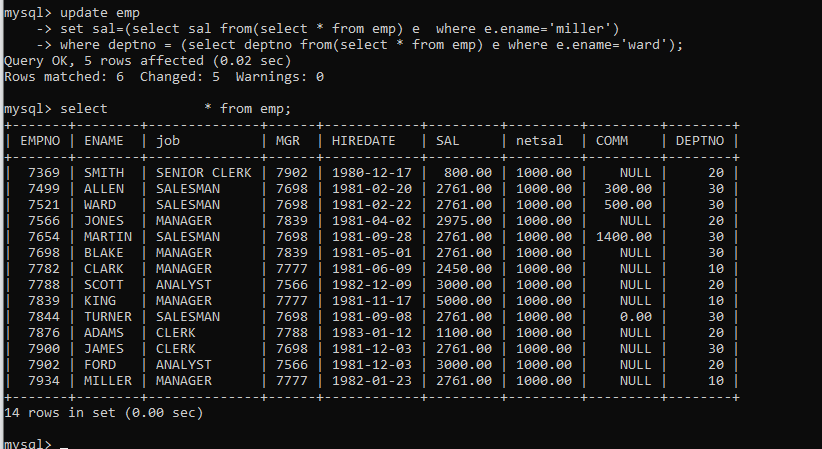
**-> set sal=(select sal from(select \* from emp) e where e.ename='miller')**

**-> where deptno = (select deptno from(select \* from emp) e where e.ename='ward');**

**before update table**

****

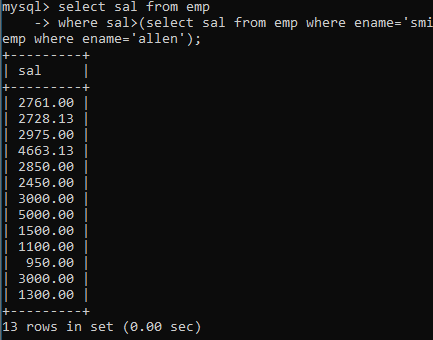
**After Update Table**

****

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

**select sal from emp**

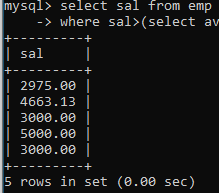
**where sal>(select sal from emp where ename='smith') or sal>(select sal from emp where ename='allen');**

****

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

**select sal from emp**

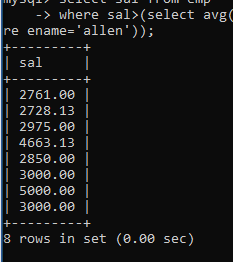
**where sal>(select avg(sal) from emp where deptno=10);**



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

**select sal from emp**

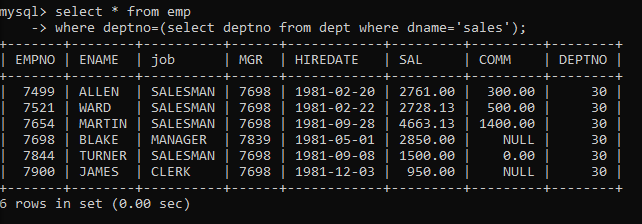
**where sal>(select avg(sal) from emp where deptno=(select deptno from emp where ename='allen'));**

****

47. list all employees who are working in sales department

**select \* from emp**

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

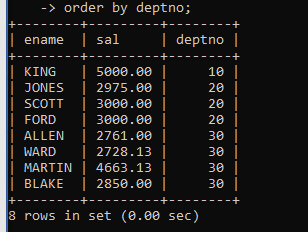
****

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

**select e.ename, e.sal, e.deptno from emp e**

**where sal > (select avg(sal) from emp where e.deptno = deptno)**

**order by deptno;**

****

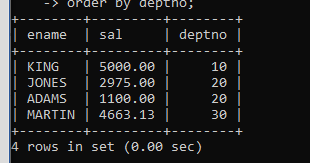
49. list all employees who earn sal < than their managers salary

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

**select e.ename, e.sal, e.deptno from emp e**

**where sal > (select avg(sal) from emp where e.job = job)**

**order by deptno;**

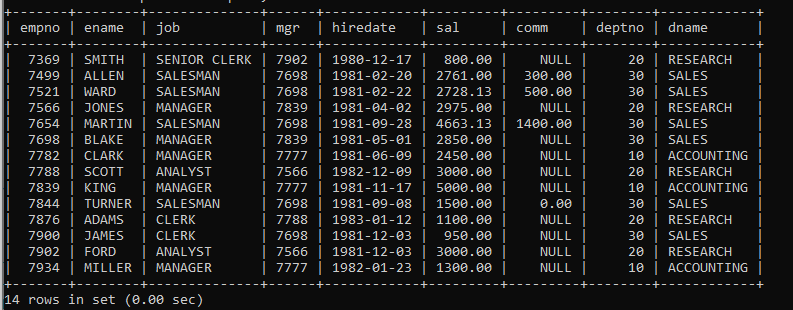


51. display employee name and department name

**select e.empno, e.ename, e.job, e.mgr, e.hiredate, e.sal, e.comm, e.deptno, d.dname**

**from emp e, dept d**

**where e.deptno = d.deptno;**



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

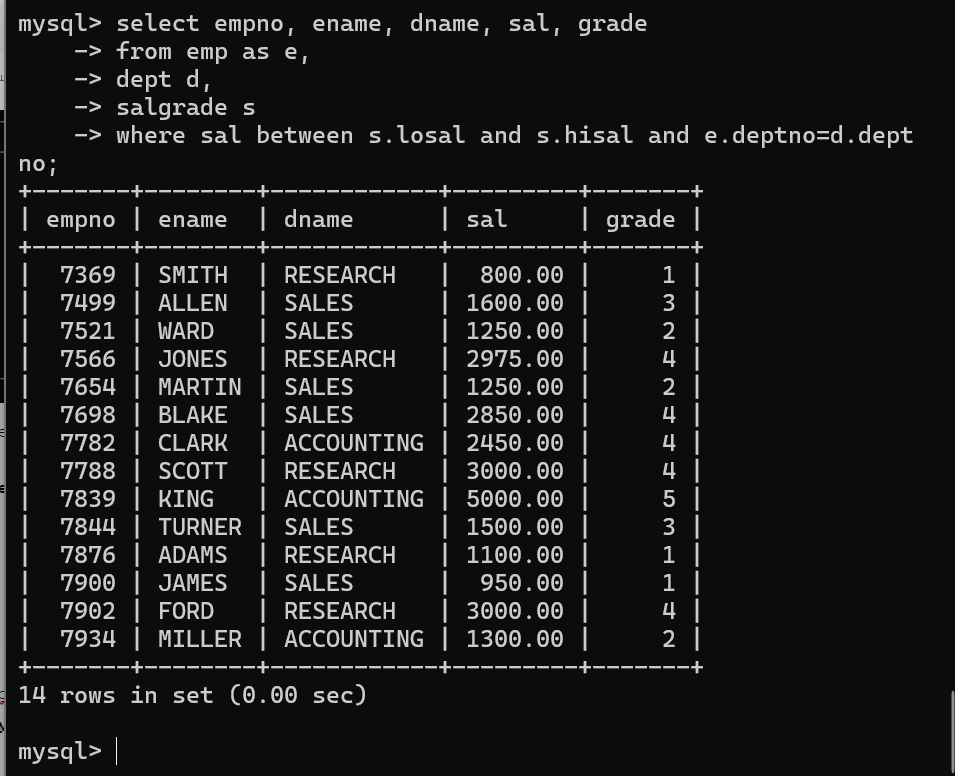
**select empno, ename, dname, sal, grade**

**from emp as e,**

**dept d,**

**salgrade s**

**where sal between s.losal and s.hisal and e.deptno=d.deptno;**

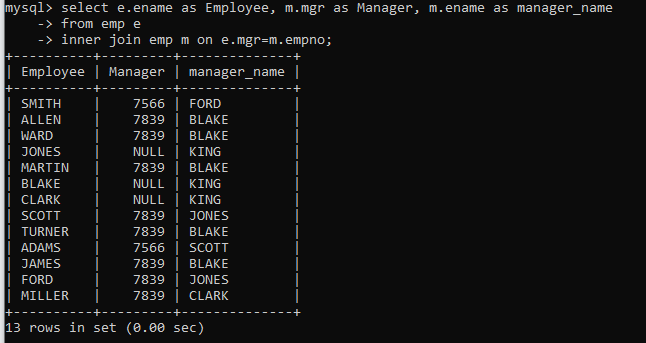
****

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

**select e.ename as Employee, m.mgr as Manager, m.ename as manager\_name**

**-> from emp e**

**-> inner join emp m on e.mgr=m.empno;**

****

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

foreign keys are marked in green

product(pid,pname,qty,price,catid,sid)

salesman (sid,sname,city)

category(cid,cnam,descritpion)

**//Salesman Table**

create table salesman(

sid int primary key,

sname varchar(20),

city varchar(50)

);

**//Category table**

create table category(

cid int primary key,

cname varchar(30),

description varchar(50)

);

**//Product Table**

create table product(

pid int primary key,

pname varchar(20),

qty int,

price int,

catid int,

sid int,

constraint fk\_cid foreign key (cid) references category(cid)

on delete set null

on update cascade,

constraint fk\_sid foreign key (sid) references salesman(sid)

on delete set null

on update cascade

);

**Product Table**

| **pid** | **pname** | **qty** | **price** | **cid** | **sid** |
| --- | --- | --- | --- | --- | --- |
| 111 | pepsi | 40 | 50 | 4 | 11 |
| 122 | pringles | 40 | 60 | 1 | 14 |
| 123 | lays | 30 | 30 | 1 | 12 |
| 124 | dairy milk | 40 | 60 | 2 | 14 |
| 134 | nachos | 50 | 50 | 1 | 12 |

**salesman Table**

| **sid** | **sname** | **address** |
| --- | --- | --- |
| 101 |  |  |
| 102 |  |  |

**category Table**

| **cid** | **cname** | **description** |
| --- | --- | --- |
| 10 |  |  |
| 20 |  |  |

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 following tables and solve following questions(primary keys are marked in yellow)foreign keys are marked in green**

faculty(fid,fname,spskill1,spskill2)

courses(cid,cname,rid,fid)

room(roomid,rname,rloc)

**//Faculty**

create table faculty(

fid int primary key,

fname varchar(20),

spskill1 varchar(2),

spskill2 varchar(2)

);

**//courses**

create table courses(

cid int primary key,

cname varchar(20),

rid int,

fid int,

constraint fk\_rid foreign key(rid) references room(roomid),

constraint fk\_fid foreign key(fid) references faculty(fid)

);

**//room**

create table room(

roomid int primary key,

rname varchar(20),

rloc varchar(10)

);

faculty

fid fname spskill1 spskill2

10 kjzhcjhz a b

11 sdd x z

12 lksjk a x

13 ksdjlkj a b

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

insert into faculty values(11,'sdd','x','z');

insert into faculty values(12,'lksjk','a','x');

insert into faculty values(13,'ksdjlkj','a','b');

courses

cid cname rid fid

121 DBDA 100 10

131 DAC 101

141 DTISS

151 DIOT 105 12

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

insert into courses values(131,'DAC',101, null);

insert into courses values(141,'DITISS',null,null);

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

Room

roomid rname rloc

100 jasmin 1st floor

101 Rose 2nd floor

105 Lotus 1st floor

103 Mogra 1st floor

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

insert into room values(101,'Rose','2st floor');

insert into room values(105,'Lotus','1st floor');

insert into room values(103,'Mogra','1st floor');

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

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

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

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

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

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

create table product(  
qty int

);

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

**PRODUCT TABLE**

**qty >0, default 20.00,pname not null and unique**

create table product(

prodid int primary key,

pname varchar(20) not null unique,

qty int check(qty>0) default 20,

price float(7,2),

catid int,

sid int,

constraint fik\_catid foreign key(catid) references category(cid)

on delete set null

on update cascade,

constraint fik\_sid foreign key(sid) references salesman(sid)

on delete set null

on update cascade

);

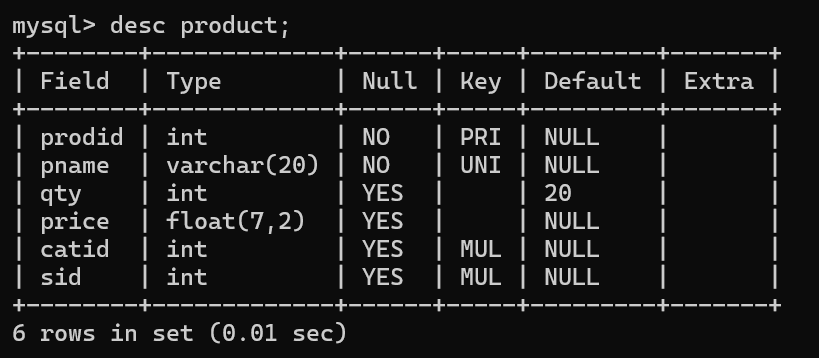
insert into product values(123,'lays',30,30,1,12);

insert into product values(111,'pepsi',40,50,4,11);

insert into product values(134,'nachos',50,50,1,12);

insert into product values(124,'dairy milk',40,60,2,14);

insert into product values(122,'pringles',40,60,1,14);



**SALESMAN**

sname -----not null

create table salesman(

sid int primary key,

sname varchar(20) not null,

city varchar(20)

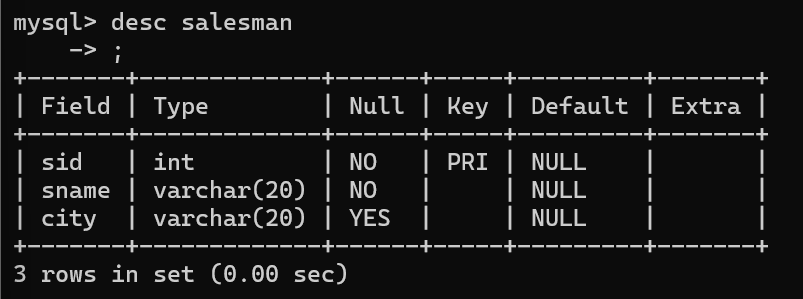
);

insert into salesman values(11, 'Rahul','Pune');

insert into salesman values(12, 'Kirti', 'Mumbai');

insert into salesman values(13, 'Prasad', 'Nashik');

insert into salesman values(14, 'Arnav', 'Amaravati');



**category**

cname unique and not null

create table category(

cid int primary key,

cname varchar(20) not null unique,

description varchar(50)

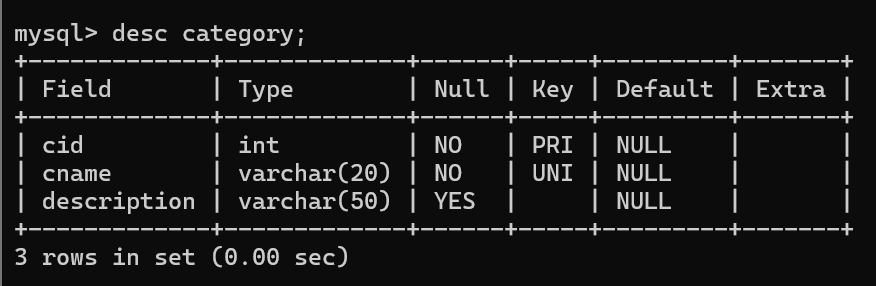
);

insert into category values(1, 'chips', 'very crunchy');

insert into category values(2, 'chocolate', 'very chocolaty');

insert into category values(3, 'snacks', 'yummy');

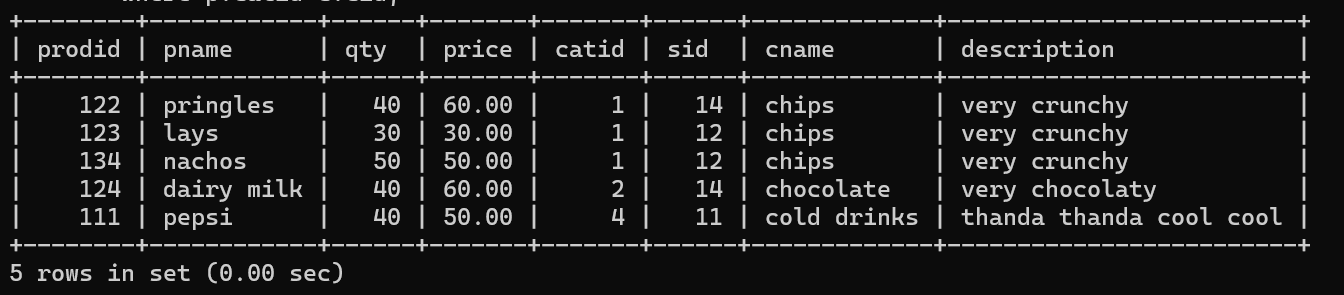
insert into category values(4, 'cold drinks' , 'thanda thanda cool cool');



1. List all products with category chips

**select prodid, pname, qty, price, catid, sid, cname, description from product as p, category as c**

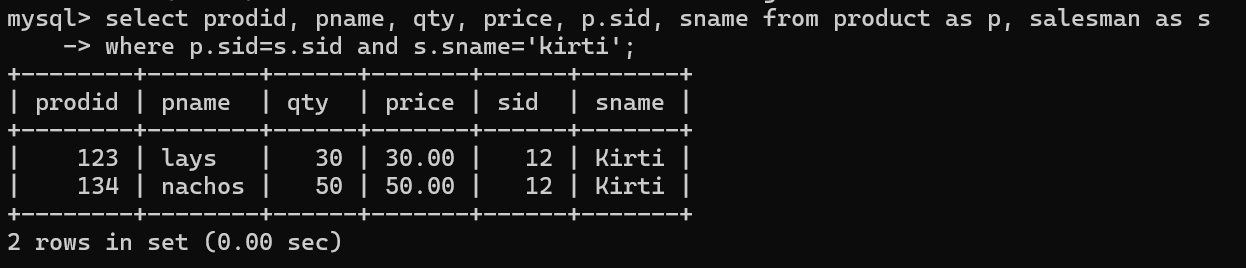
**where p.catid=c.cid;**



2. display all products sold by kirti

**select prodid, pname, qty, price, p.sid, sname from product as p, salesman as s**

**where p.sid=s.sid and s.sname='kirti';**

****

3. 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