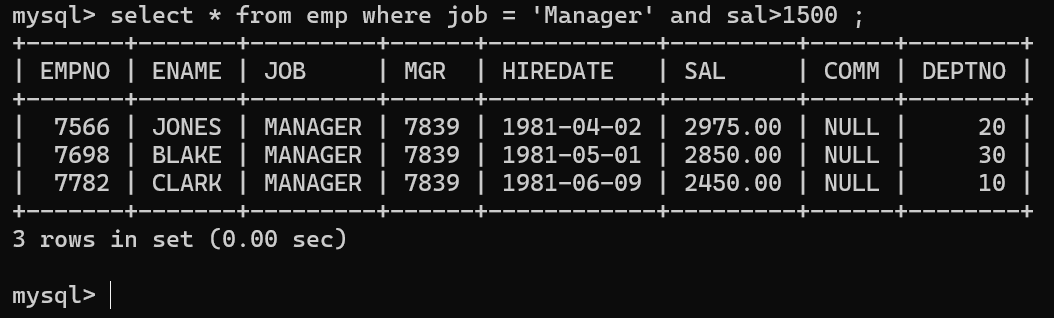
**Practice DQL statement**

**Write SQL statement for the following**

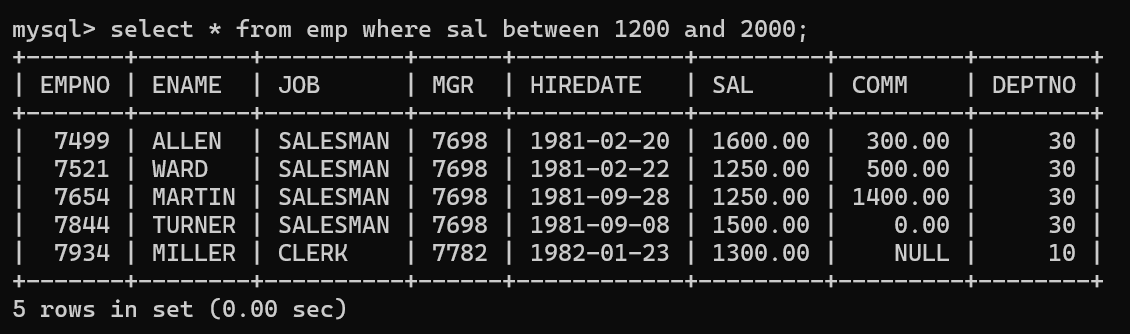
**1. To find all managers with salary >1500**

select \* from emp where job = 'Manager' and sal>1500 ;



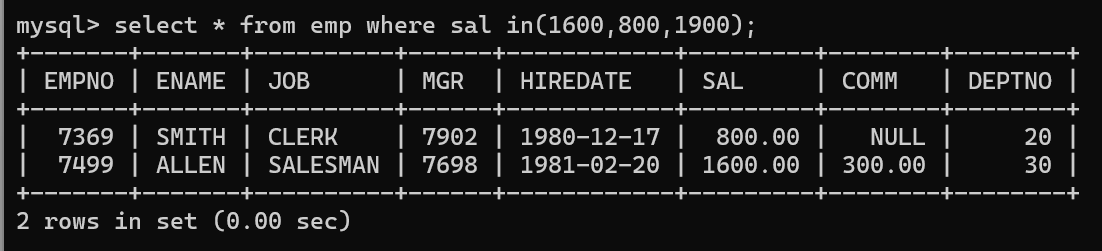
**2. list all employees with sal >1200 and < 2000**

select \* from emp where sal between 1200 and 2000;



**3. list all employees with sal is 1600 or sal is 800 or sal is 1900**

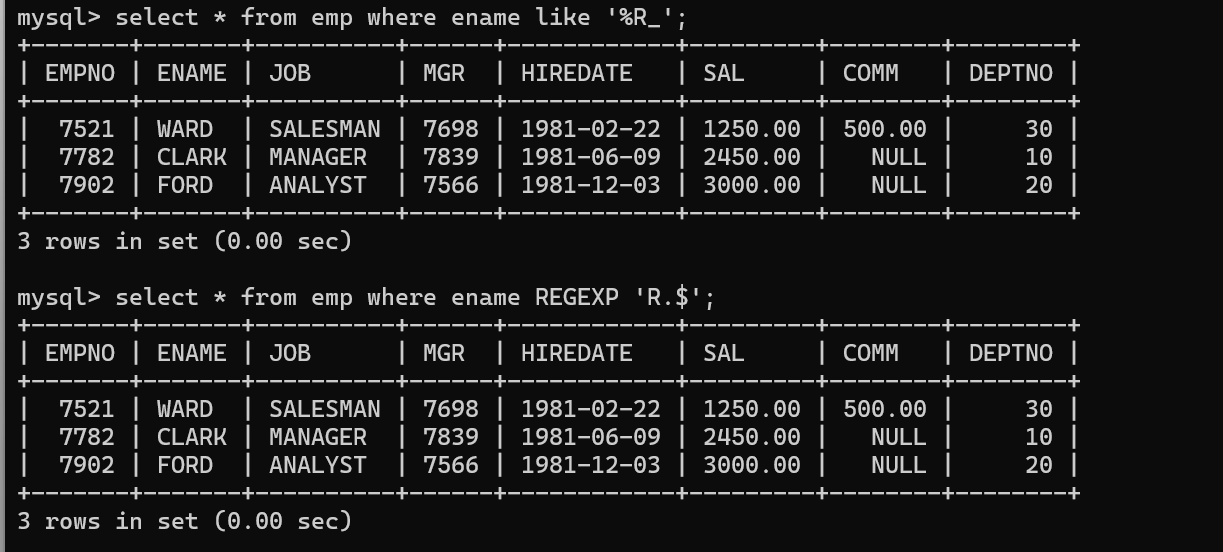
select \* from emp where sal in(1600,800,1900);



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

select \* from emp where ename like '%R\_';

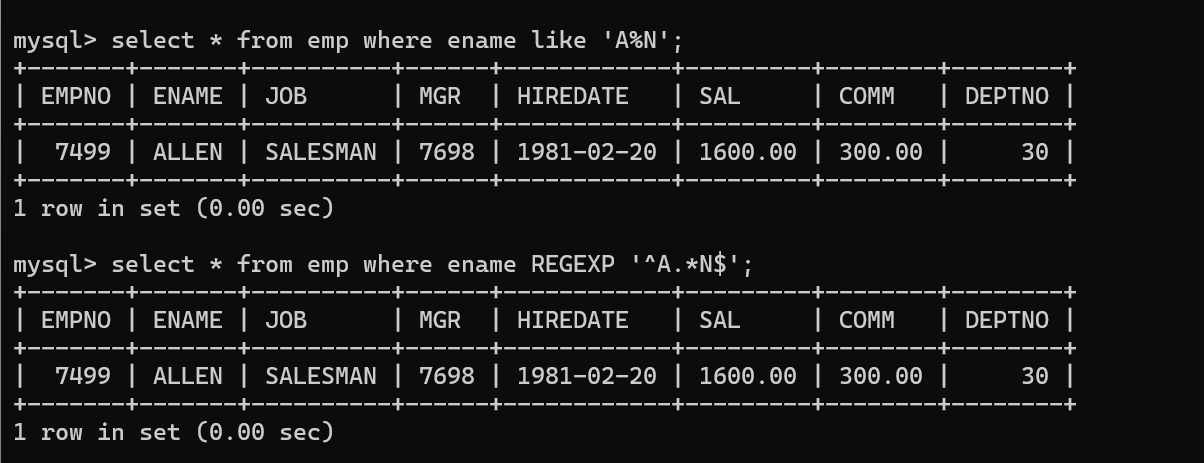
select \* from emp where ename REGEXP 'R.$';



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

select \* from emp where ename like 'A%N';

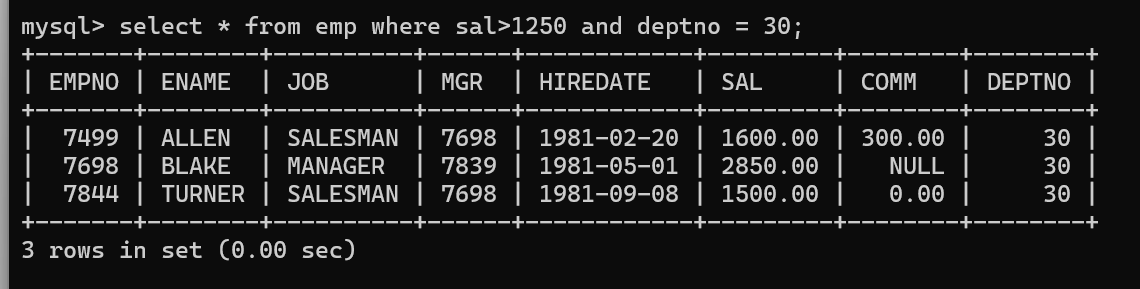
select \* from emp where ename REGEXP '^A.\*N$';



**Q2. Solve following**

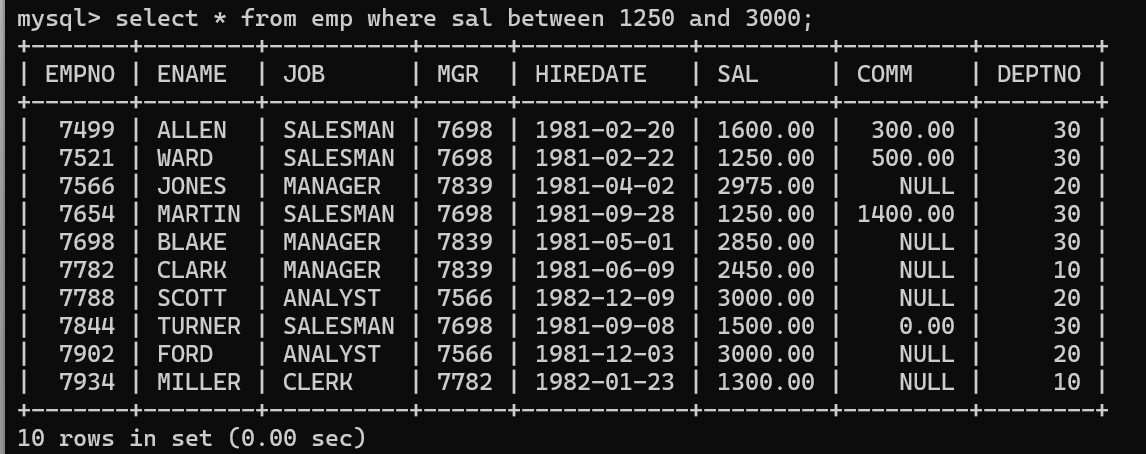
1. **list all employees with salary > 1250 and dept no=30**

select \* from emp where sal>1250 and deptno = 30;



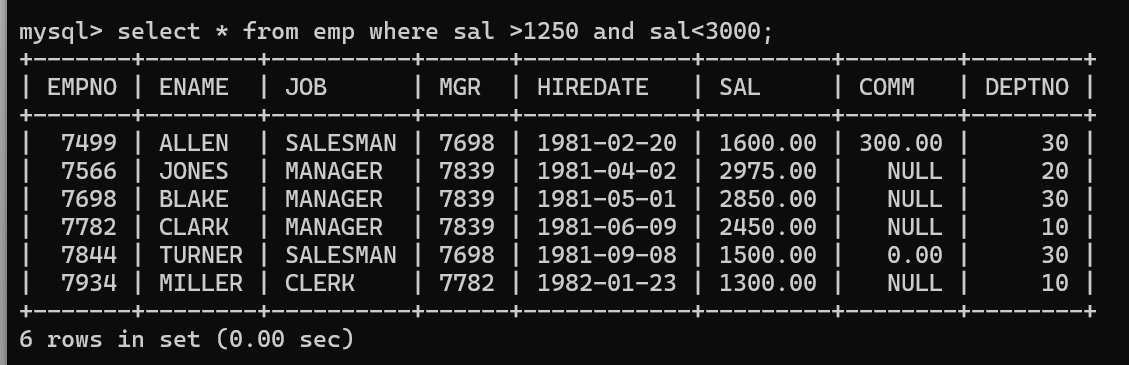
1. **list all employees with salary >=1250 and <= 3000**

select \* from emp where sal between 1250 and 3000;



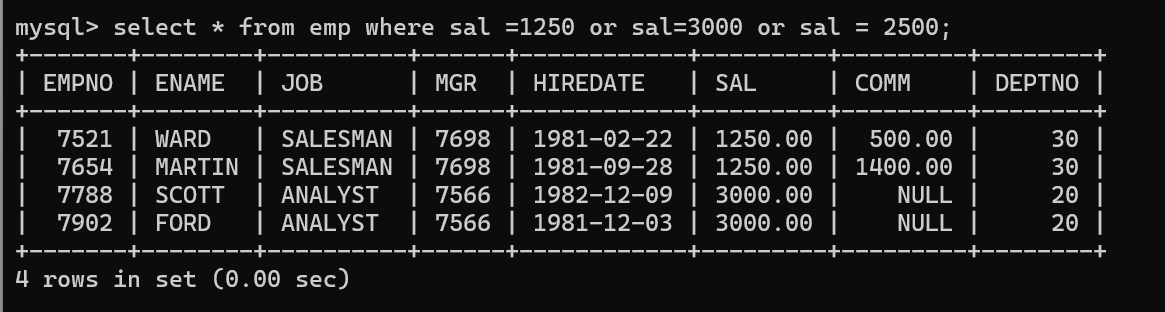
1. **list all employees with salary >1250 and < 3000**

select \* from emp where sal >1250 and sal<3000;



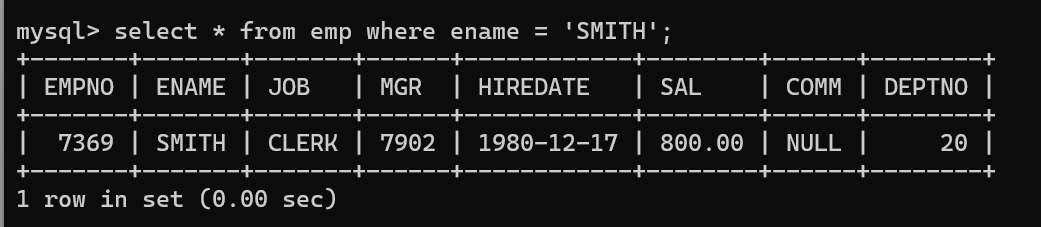
1. **list all employees with salary either equal to 3000 or 1250 or 2500**i

select \* from emp where sal =1250 or sal=3000 or sal = 2500;



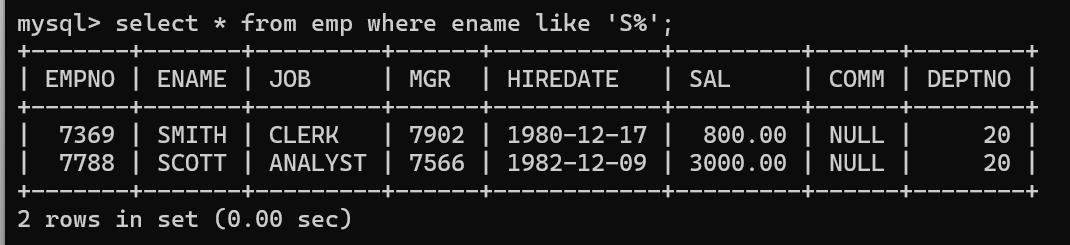
1. **list all employee with name=SMITH**

select \* from emp where ename = 'SMITH';



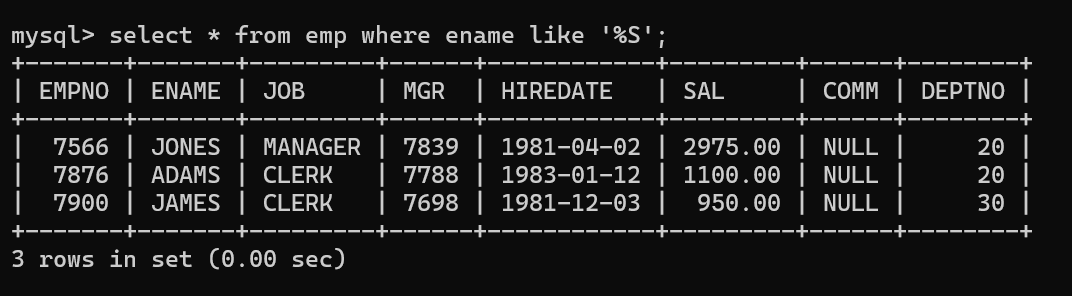
1. **list all employees with name starting with S**

select \* from emp where ename like 'S%';



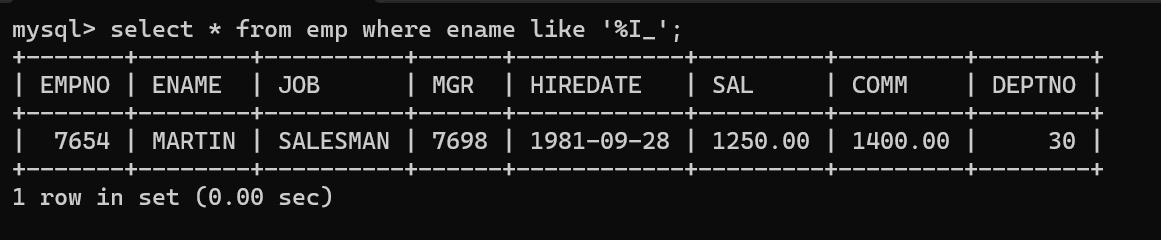
1. **list all employees with name ending with S**

select \* from emp where ename like '%S';



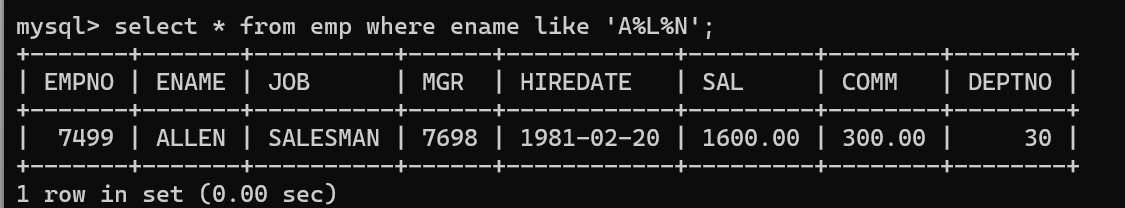
1. **list all employees with name contains I at 2nd position**

select \* from emp where ename like '%I\_';



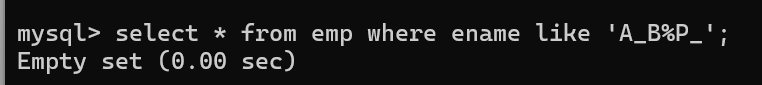
1. **list all employees with name starts with A ends with N and somewhere in between L is there**

select \* from emp where ename like 'A%L%N';



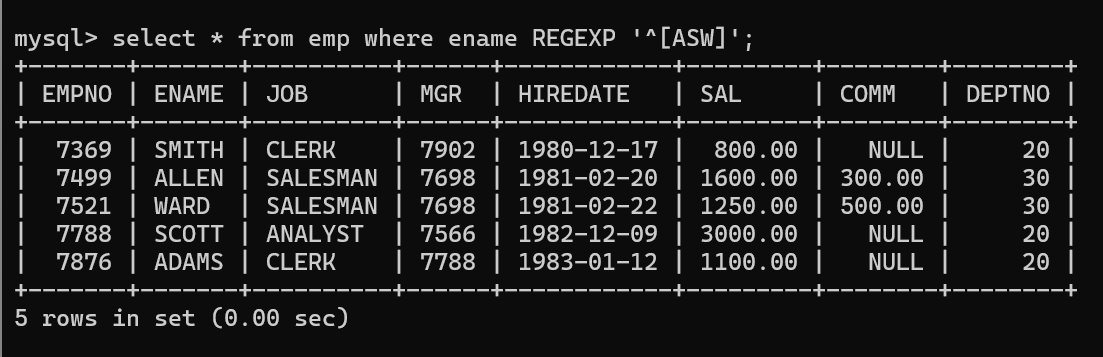
1. **list all employees with name starts with A and B at 3 rd position and P at second last position**

select \* from emp where ename like 'A\_B%P\_';



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

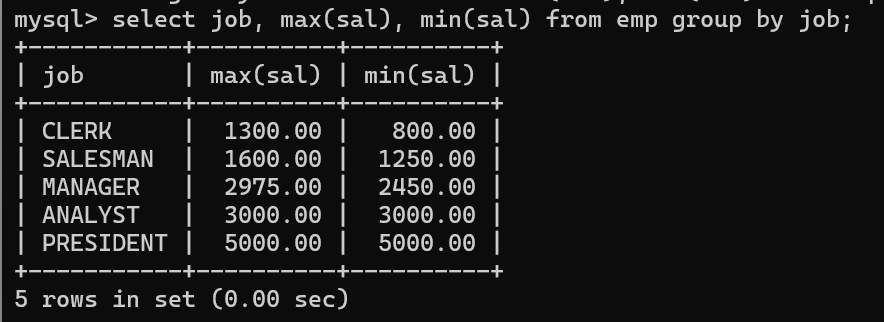
select \* from emp where ename REGEXP '^[ASW]';



**Practice Aggregate functions**

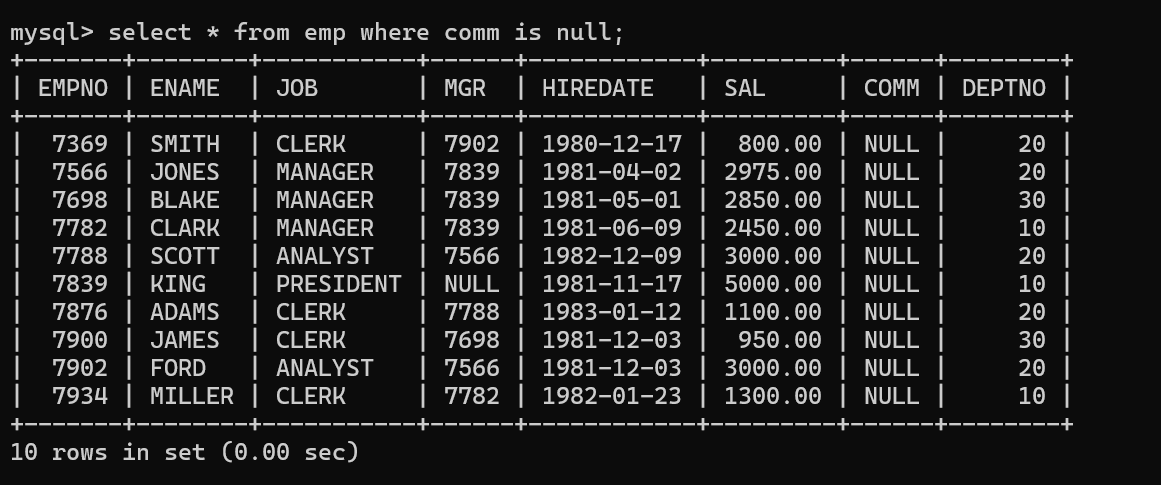
1. **find max sal and min sal for each job**

select job, max(sal), min(sal) from emp group by job;



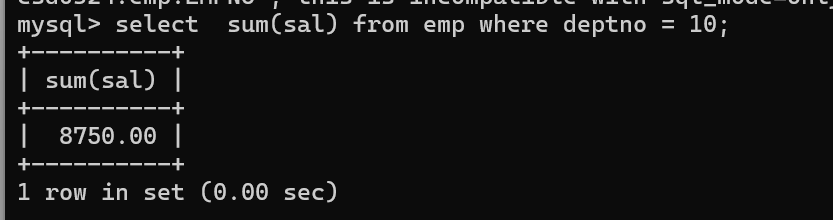
1. **find how many employess have not received commission**

select \* from emp where comm is null;



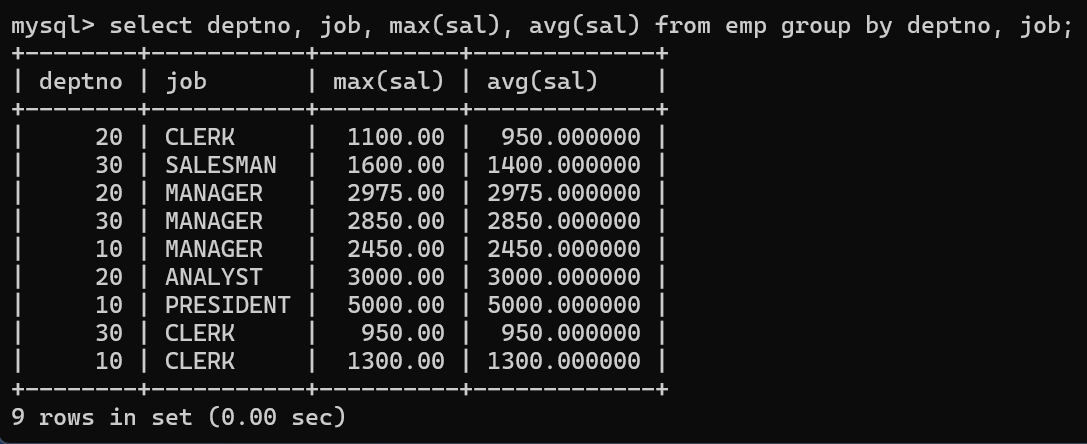
1. **find sum of sal of all employees working in dept no 10**

select sum(sal) from emp where deptno = 10;



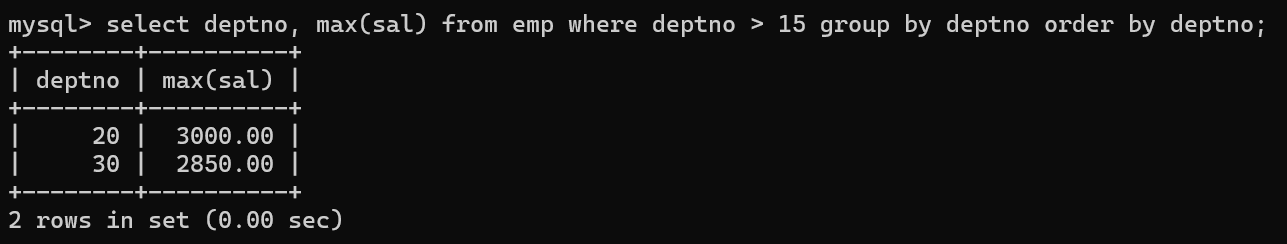
1. **find maximum salary,average sal for each job in every department**

select deptno, job, max(sal), avg(sal) from emp group by deptno, job;



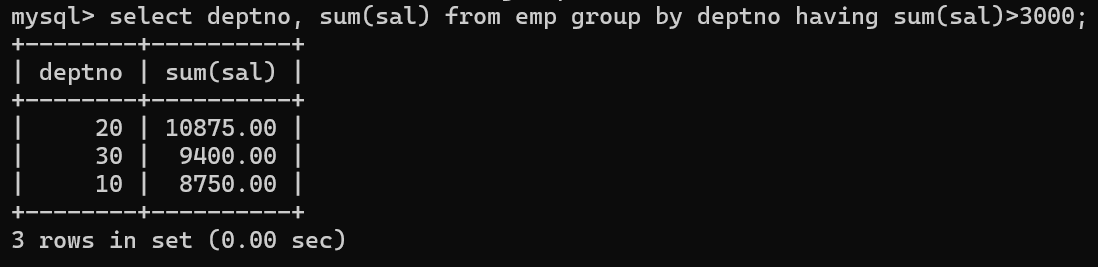
1. **find max salary for every department if deptno is > 15 and arrange data in deptno order.**

select deptno, max(sal) from emp where deptno > 15 group by deptno order by deptno;



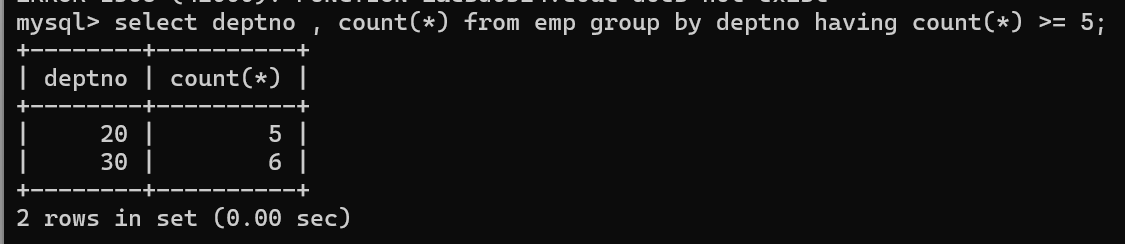
1. **find sum salary for every department if sum is > 3000**

select deptno, sum(sal) from emp group by deptno having sum(sal)>3000;



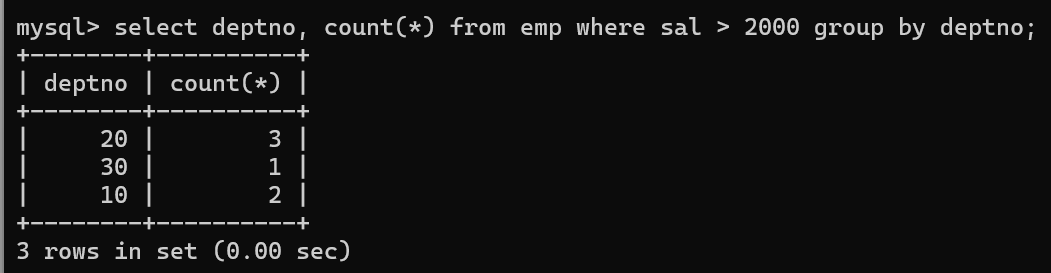
1. **list all department which has minimum 5 employees**

select deptno , count(\*) from emp group by deptno having count(\*) >= 5;



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

select deptno, count(\*) from emp where sal > 2000 group by deptno;



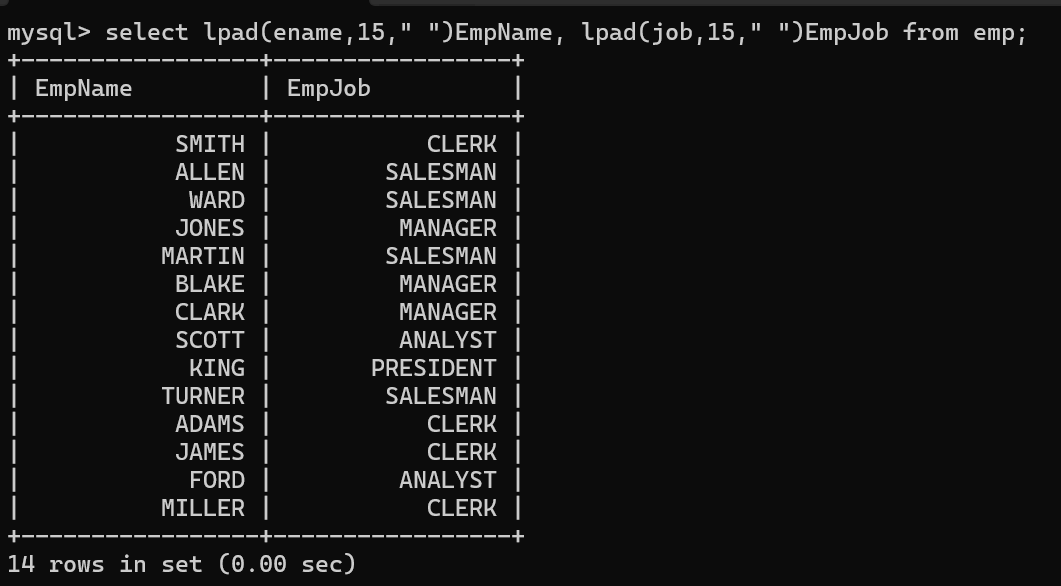
1. **list all enames and jobs in small case letter**

select lower(ename) EmpName, lower(job)Job from emp;



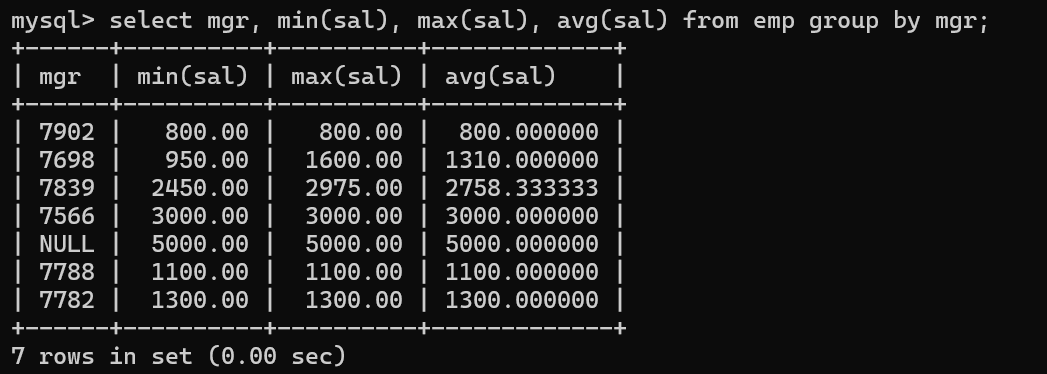
1. **list all names and jobs so that the length of name should be 15 if it is smaller then add spaces to left**

select lpad(ename,15," ")EmpName, lpad(job,15," ")EmpJob from emp;



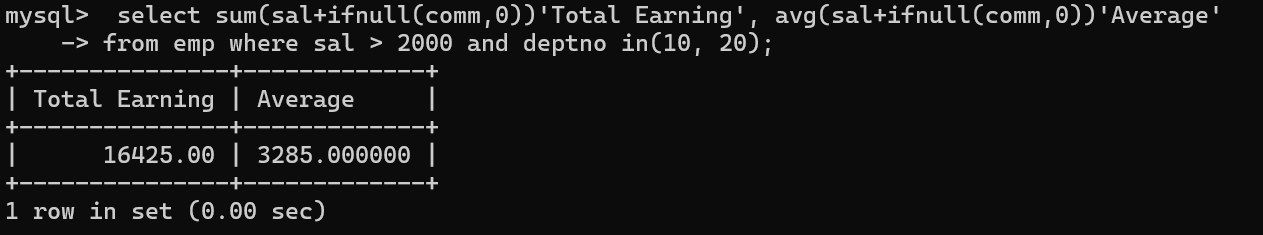
1. **display min sal,max sal, average sal for all employees working under same manager**

select mgr, min(sal), max(sal), avg(sal) from emp group by mgr;



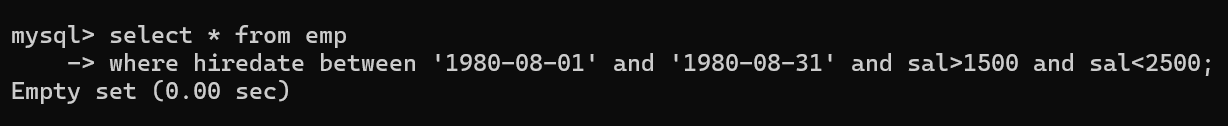
1. **find sum of total earnings(sal+comm), average of sal+comm, for all employees who earn sal > 2000 and work in either dept no 10 or 20**

select sum(sal+ifnull(comm,0))'Total Earning', avg(sal+ifnull(comm,0))'Average' from emp where sal > 2000 and deptno in(10, 20);



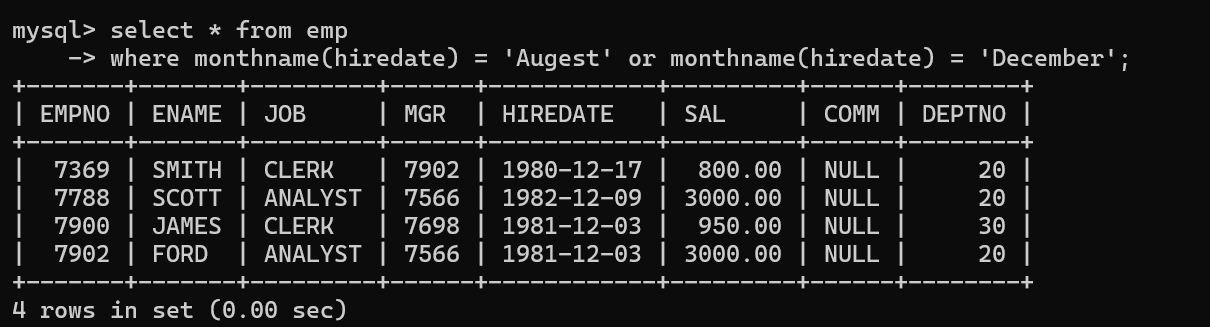
1. **list all employees who joined in Aug 1980 and salary is >1500 and < 2500**

select \* from emp where hiredate between '1980-08-01' and '1980-08-31' and sal>1500 and sal<2500;



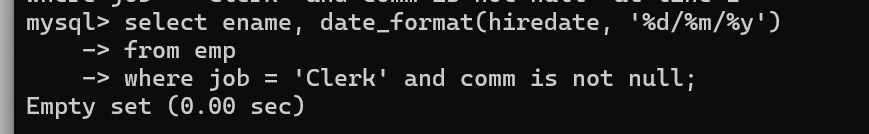
1. **list all employees joined in either aug or may or dec**

select \* from emp where monthname(hiredate) = 'Augest' or monthname(hiredate) = 'December';



1. **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') from emp where job = 'Clerk' and comm is not null;



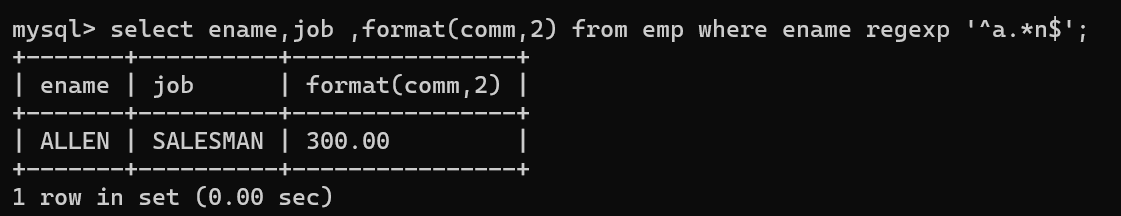
1. **list empcode,empno,name and job for each employee. (note :empcode is 3 to 5 characters from name and last 2 characters of job)**

select concat(left(ename,4) ,right(job,2))empcode, ename, job from emp;

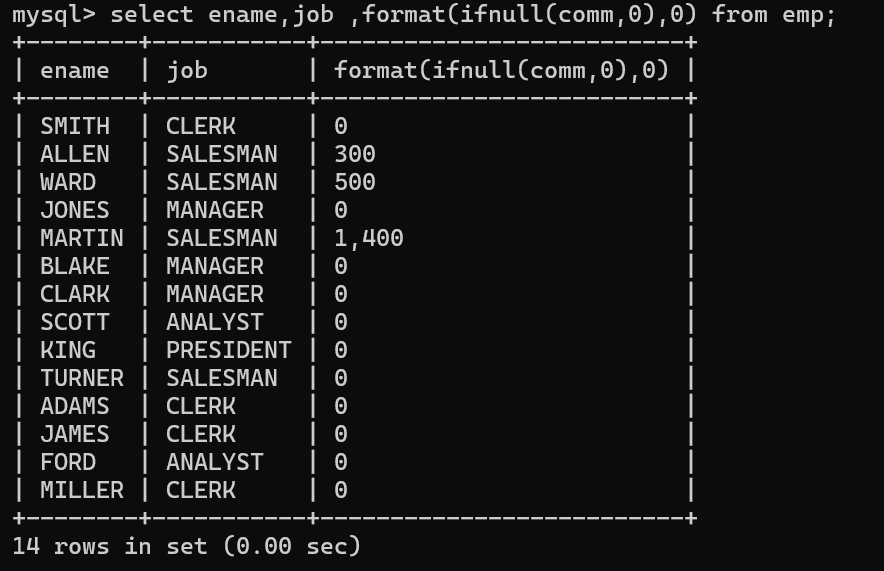


1. **display thousand separator and $ symbol for commission if it is null then display it as 0 for all employees whose name starts with A and ends with N**

select ename,job ,format(comm,2) from emp where ename regexp '^a.\*n$';



select ename,job ,format(ifnull(comm,0),0) from emp;



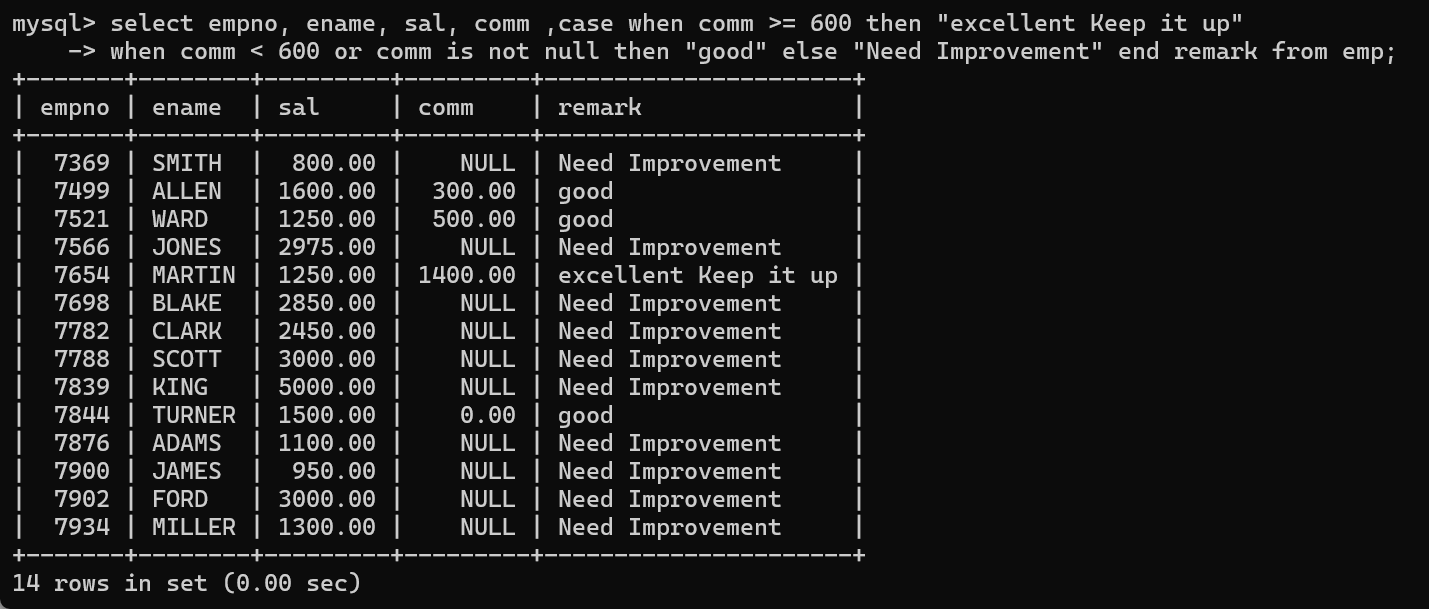
**29. Display empid,name,sal,comm,remark Remark should base on following conditions**

**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.**

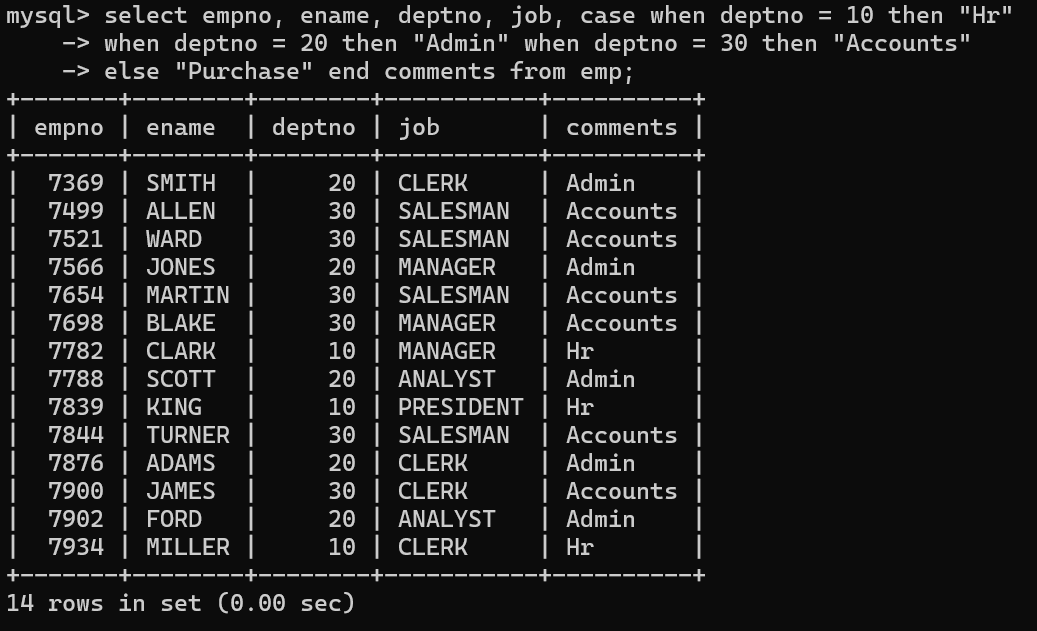
**dept 10 then "Hr"**

**if 20 then "Admin"**

**if 30 then "accounts"**

**otherwise purchase**

select empno, ename, deptno, job, case when deptno = 10 then "Hr" when deptno = 20 then "Admin" when deptno = 30 then "Accounts" else "Purchase" end comments 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)**

**)**

**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,

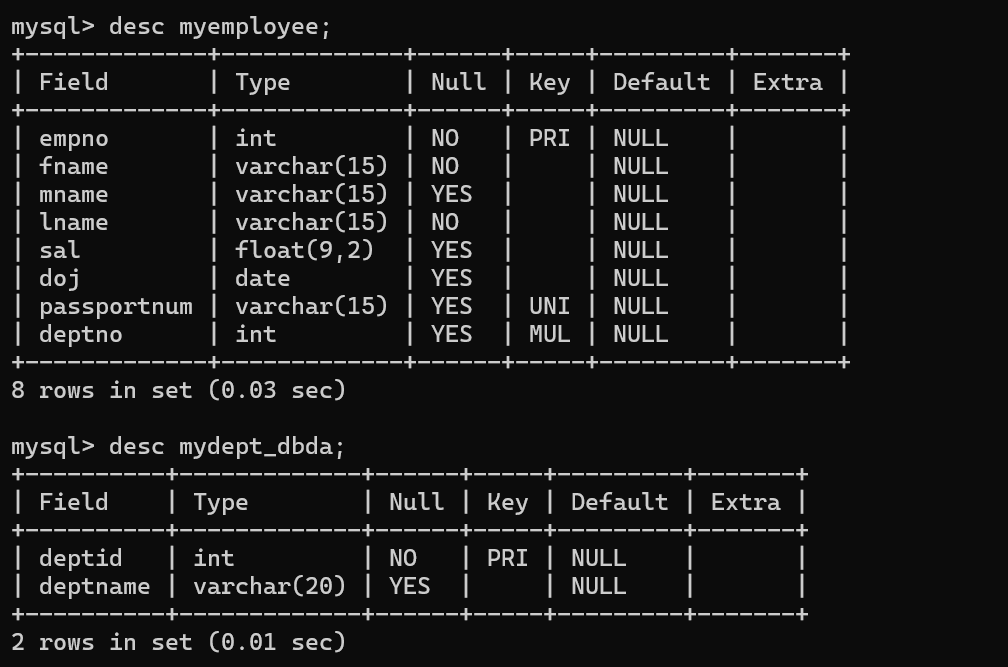
deptno int,

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)**

**)**

create table Student( sid int primary key, sname varchar(20));

create table Course(cid int primary key, cname varchar(20));

create table Marks( studid int, courseid int, markes int,

-> constraint pk primary key(studid, courseid),

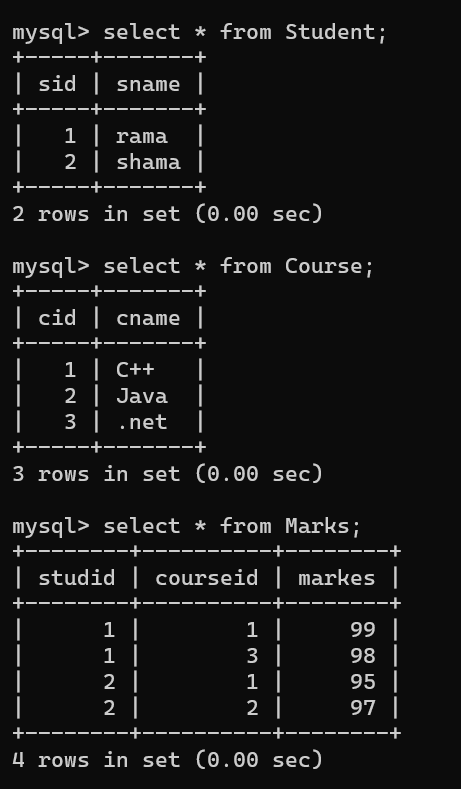
-> constraint fk\_sid foreign key(studid) references Student(sid)

-> on delete cascade on update cascade,

-> constraint fk\_cid foreign key(courseid) references Course(cid)

-> on delete cascade on update cascade

-> );



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

create table emp10 as

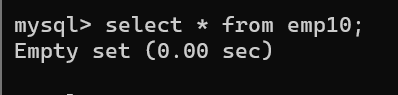
(

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 empcpy

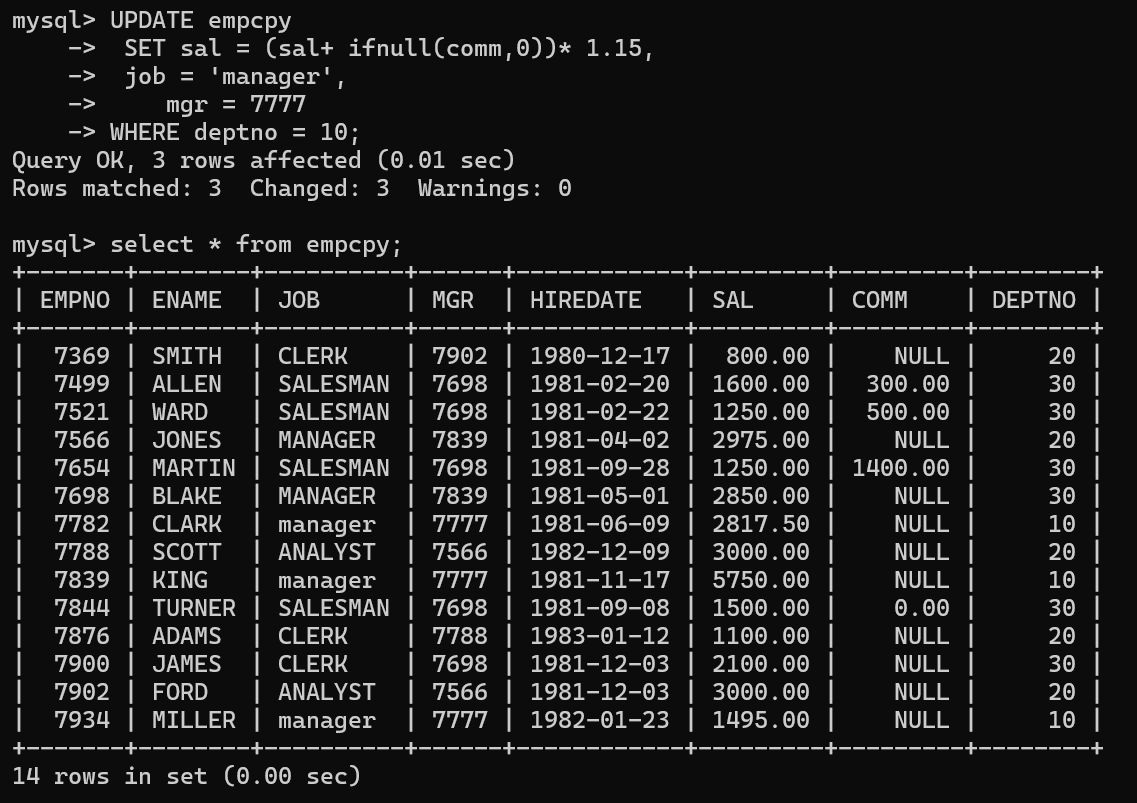
SET sal = (sal+ ifnull(comm,0))\* 1.15,

job = 'manager',

mgr = 7777

WHERE deptno = 10;

select \* from empcpy;



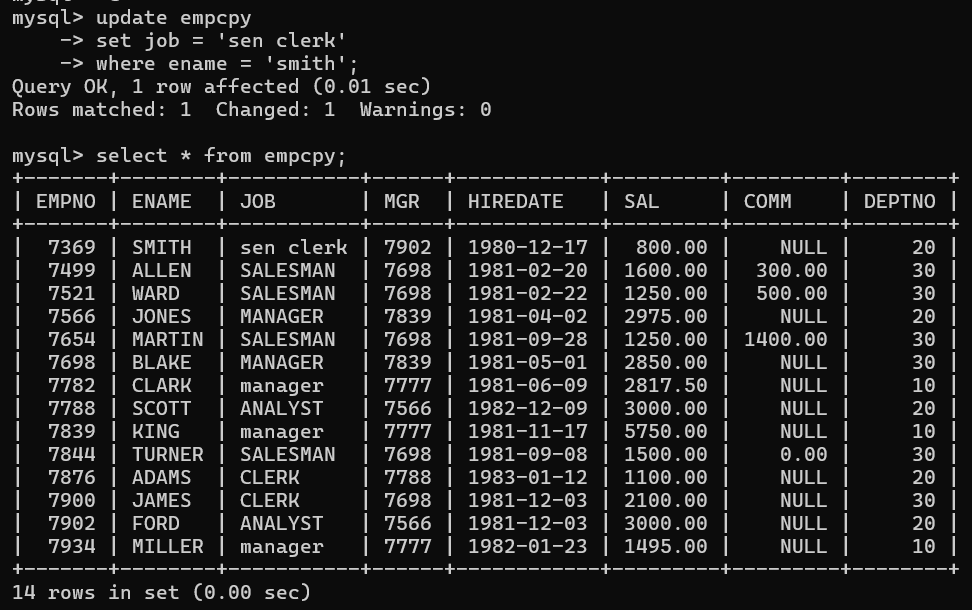
**36. change job of smith to senior clerk**

update empcpy

set job = 'sen clerk'

where ename = 'smith';

select \* from empcpy;



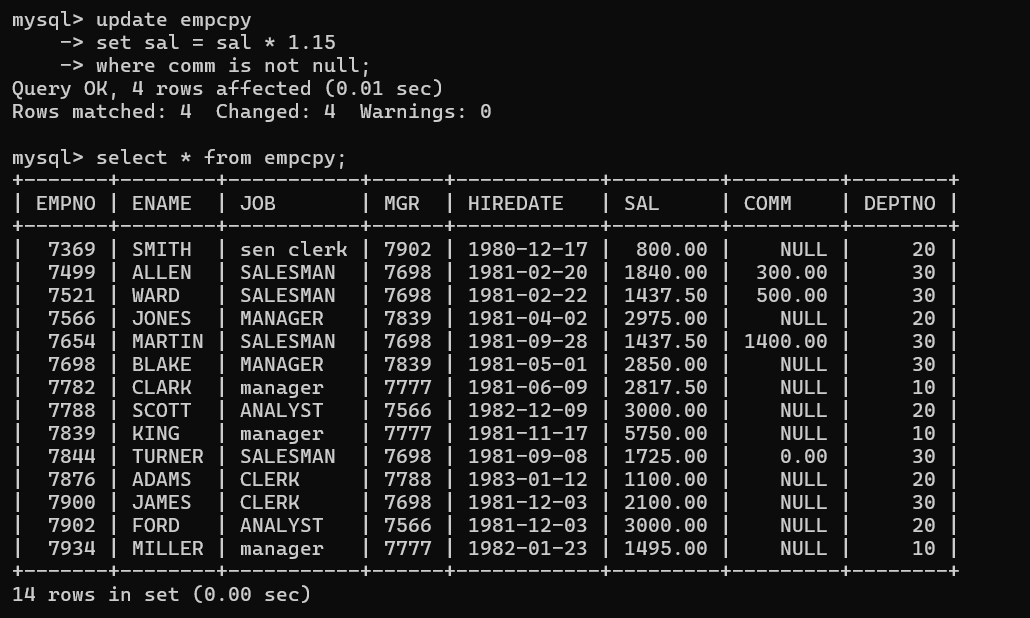
**37. increase salary of all employees by 15% if they are earning some commission**

update empcpy

set sal = sal \* 1.15

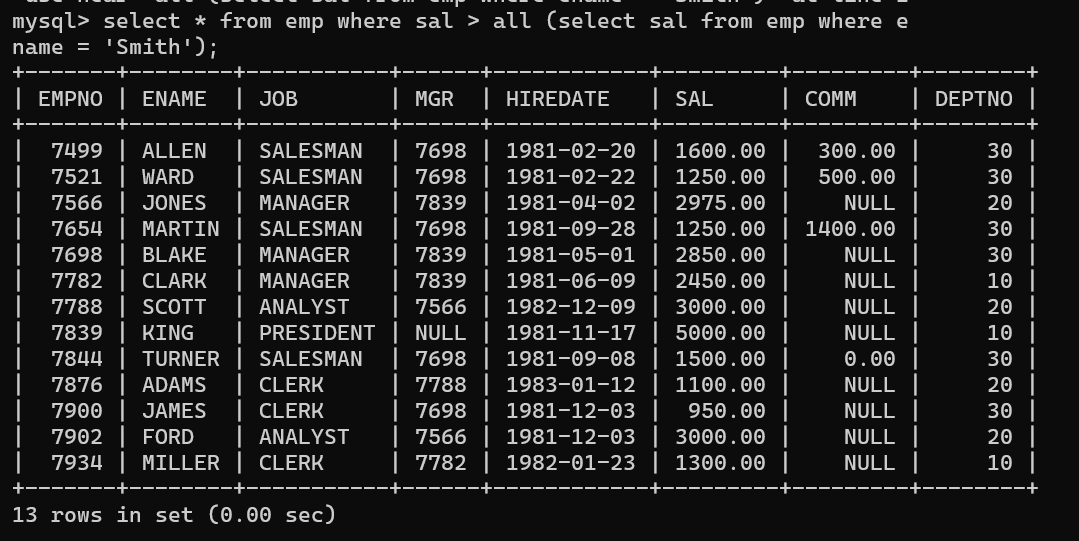
where comm is not null;

select \* from empcpy;



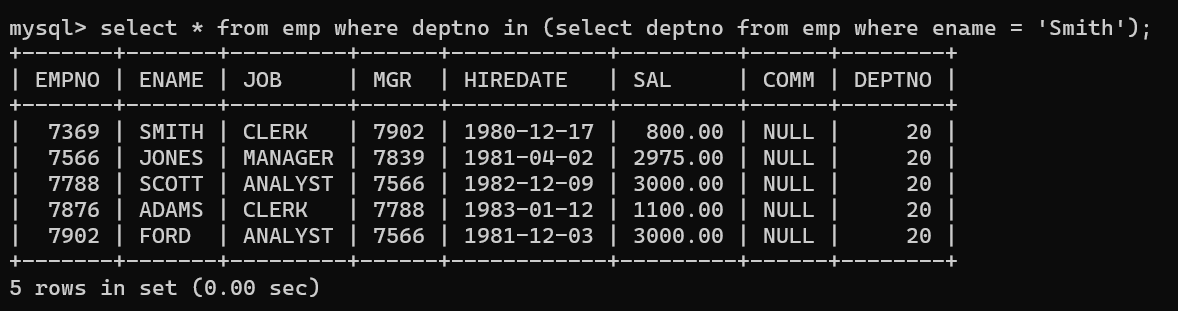
**38. list all employees with sal>smith's sal**

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

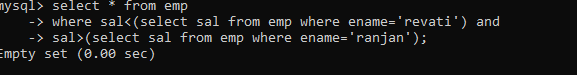
****

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

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



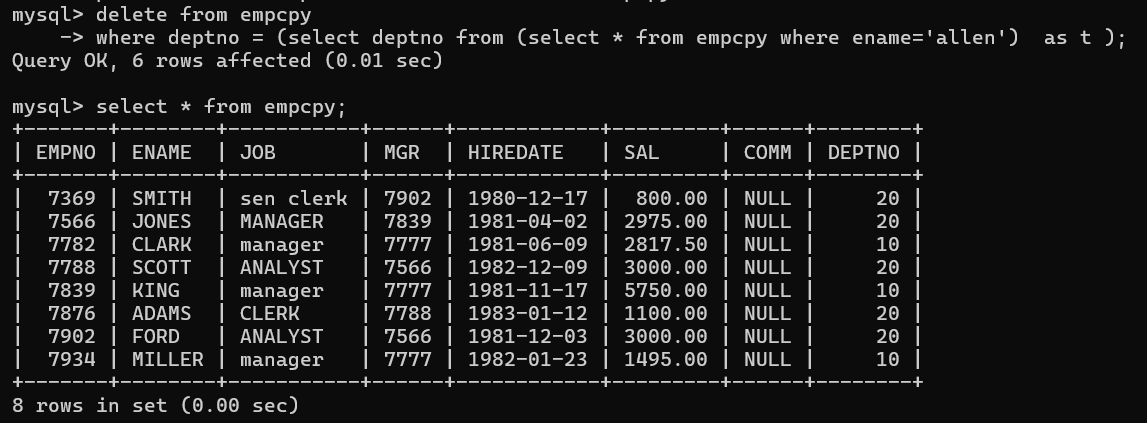
**40. list all employees with sal < rajan's sal and salary > revati's sal**



**41. delete all employees working in alan's department**

delete from empcpy

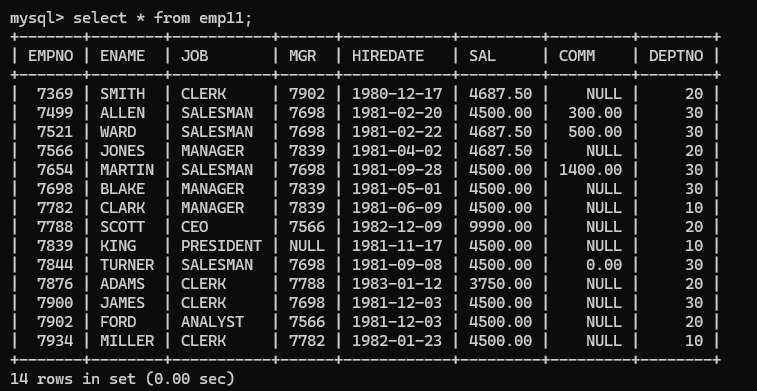
where deptno = (select deptno from (select \* from empcpy where ename='allen') as t );



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

update emp11

-> set sal = (select sal from emp where ename = "Miller") where ename = "Allen";

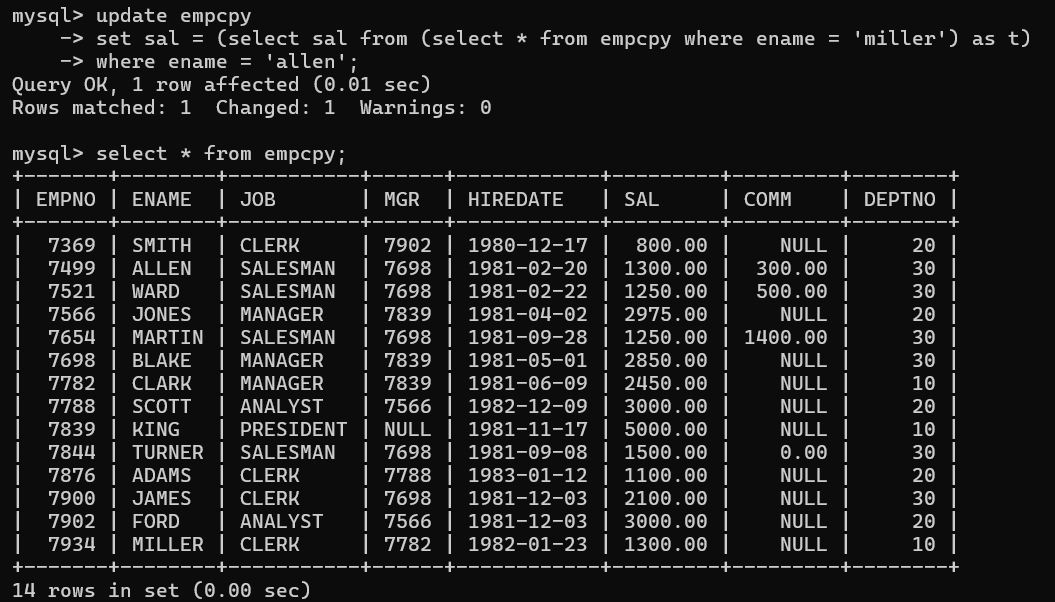
****

**43. change salary of all emplyees who working in Wall's department to the salary of Miller.**

update empcpy

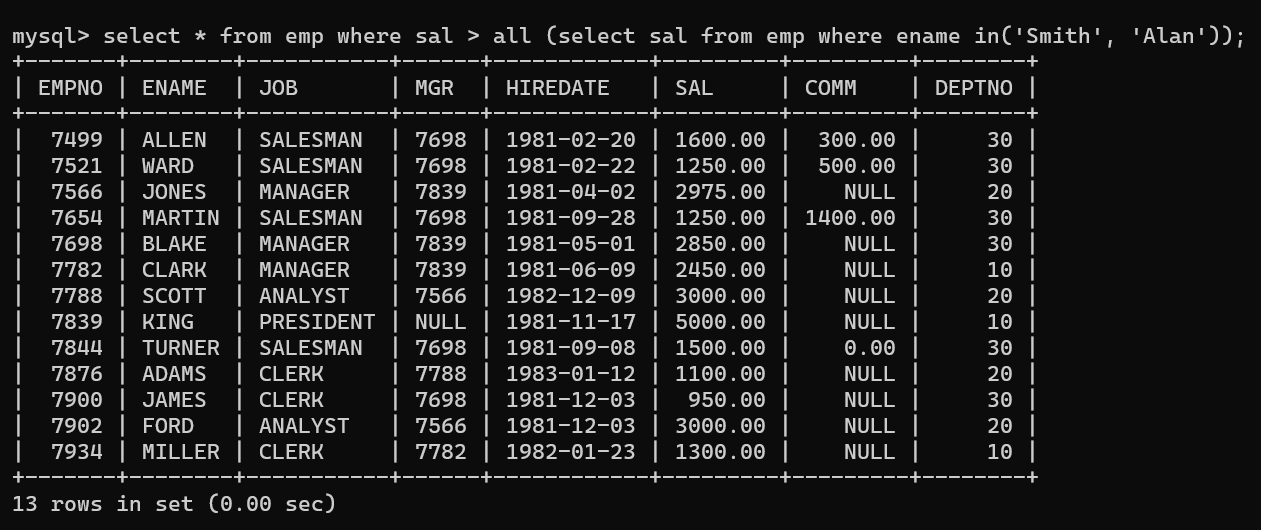
set sal = (select sal from (select \* from empcpy where ename = 'miller') as t)

where ename = 'allen';



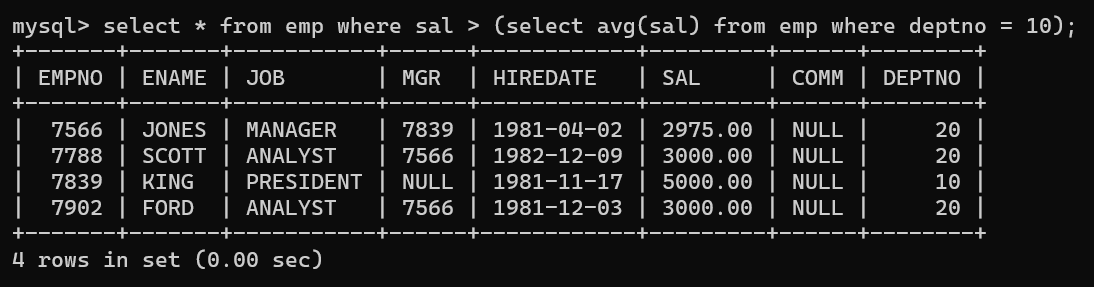
**44. list all employees with salary > either Smith's salary or alan's sal**

select \* from emp where sal > all (select sal from emp where ename in('Smith', 'Alan'));



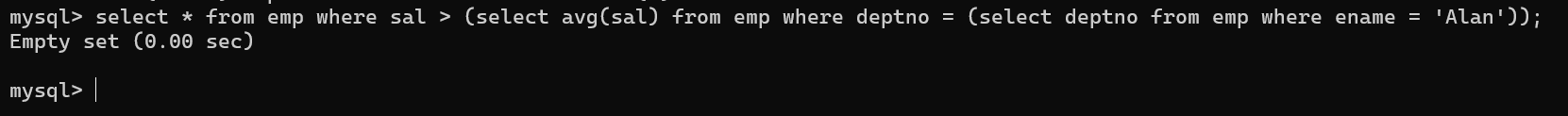
**45. list all employees who earn more than average sal of dept 10**

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

****

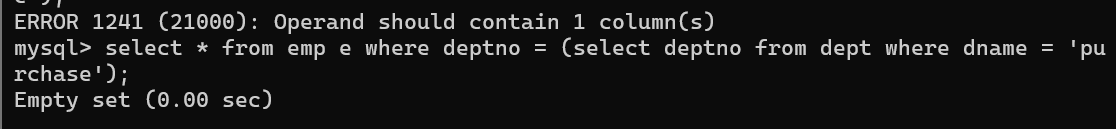
**46. list all employees who earn more than average sal of Alan's department**

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

****

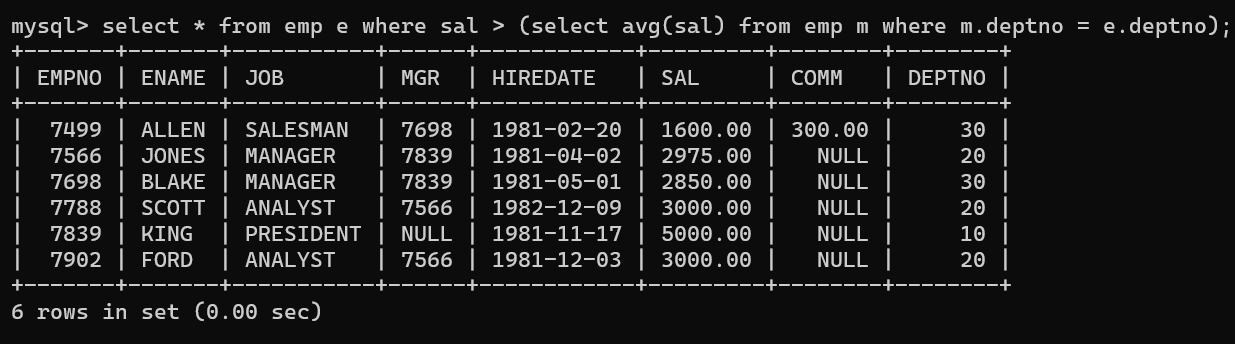
**47. list all employees who are working in purchase department**

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



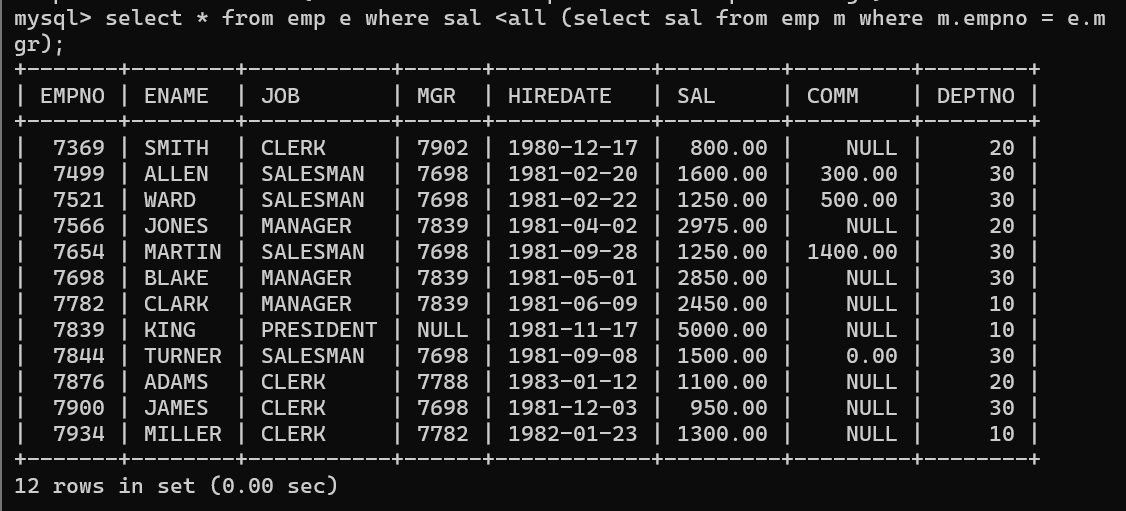
**48. list all employees who earn more than average salary of their own department**

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



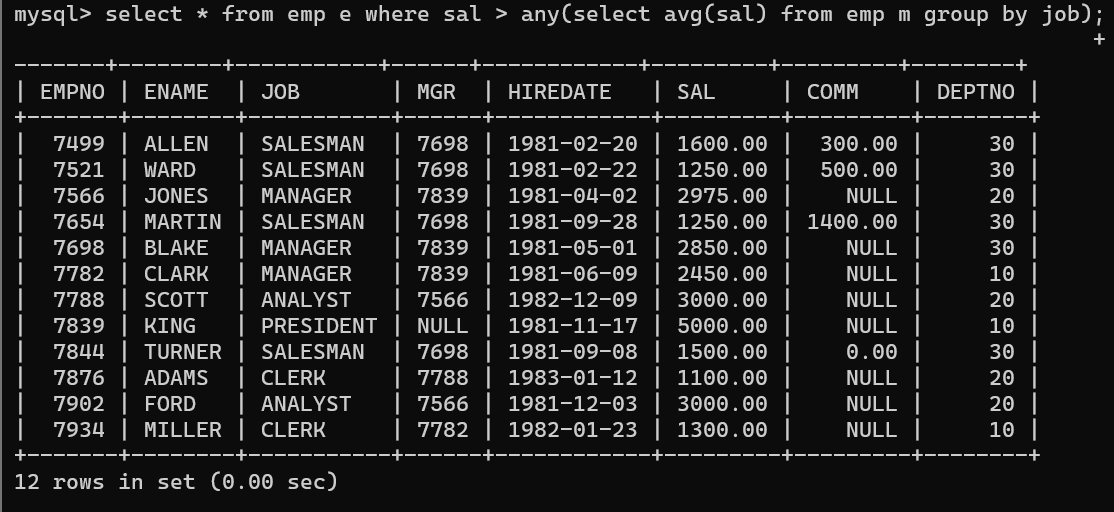
**49. list all employees who earn sal < than their managers salary**

select \* from emp e where sal <all (select sal from emp m where m.empno = e.mgr);

****

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

select \* from emp e where sal > any(select avg(sal) from emp m group by job);



**51. display employee name and department**

select ename ,dname from emp e, dept d where e.deptno = d.deptno;

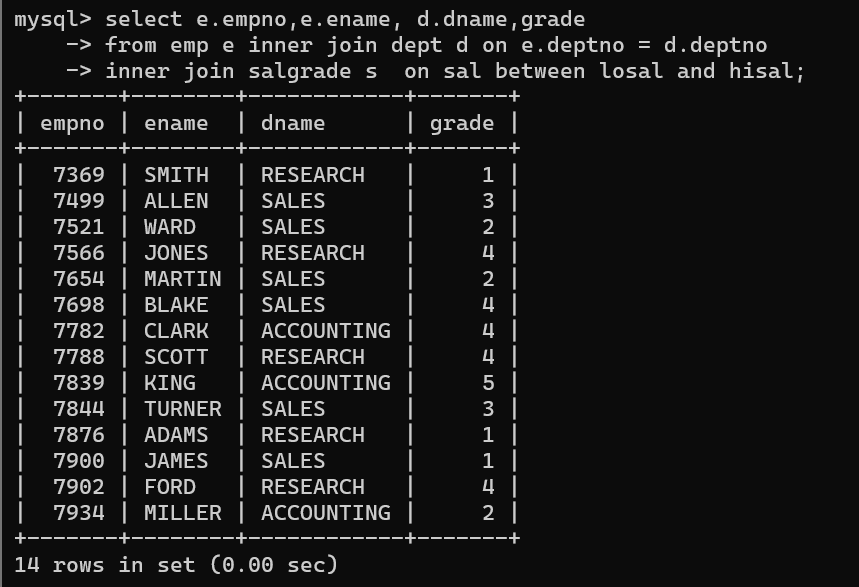
****

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

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

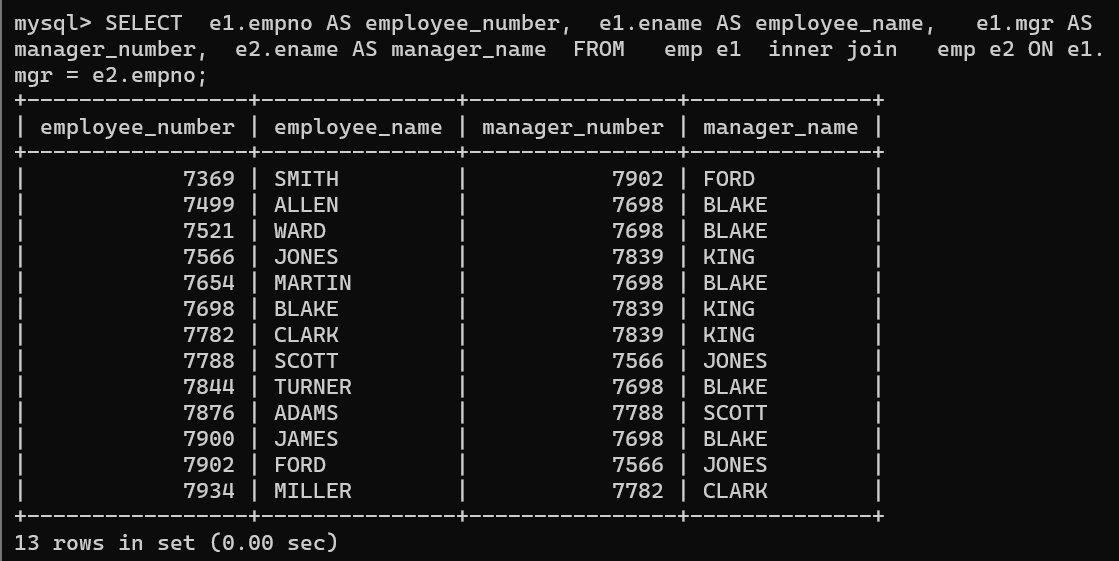
-> from emp e inner join dept d on e.deptno = d.deptno

-> inner join salgrade s on sal between losal and hisal;

****

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

SELECT e1.empno AS employee\_number, e1.ename AS employee\_name, e1.mgr AS manager\_number, e2.ename AS manager\_name FROM emp e1 inner join emp e2 ON e1.mgr = e2.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)**

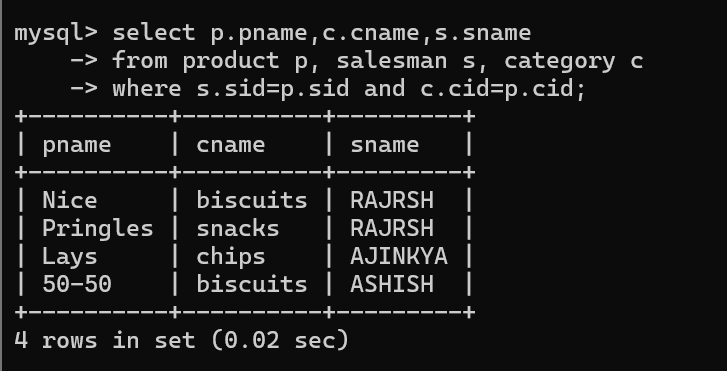
**category(cid,cnam,description)**

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

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

-> from product p, salesman s, category c

-> where s.sid=p.sid and c.cid=p.cid;

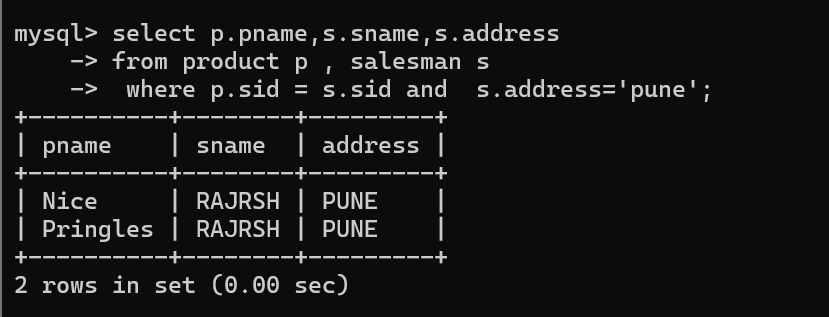


1. **list all product name and salesman name for all salesman who stays in pune**

select p.pname,s.sname,s.address

-> from product p , salesman s

-> where p.sid = s.sid and s.address='pune';

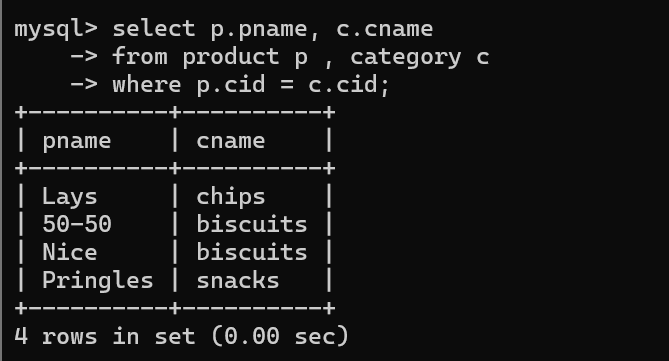


1. **list all product name and category name**

select p.pname, c.cname

from product p , category c

where 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)**

**courses11(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**

**Courses11**

**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, c.rid, r.rid, r.rname,r.location

from course11 c right join room r

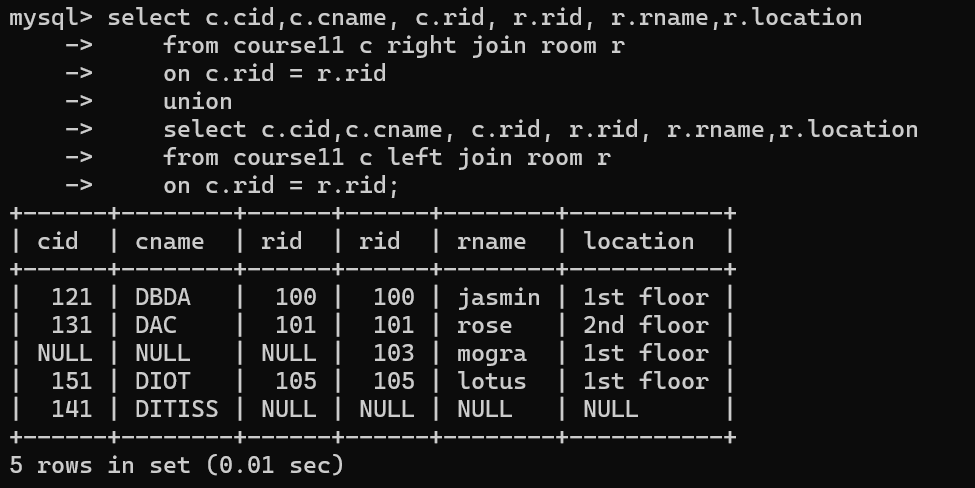
on c.rid = r.rid

union

select c.cid,c.cname, c.rid, r.rid, r.rname,r.location

from course11 c left join room r

on c.rid = r.rid;



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

select f.fid, f.fname, f.skill\_1,f.skill\_2

from faculty f right join course11 c

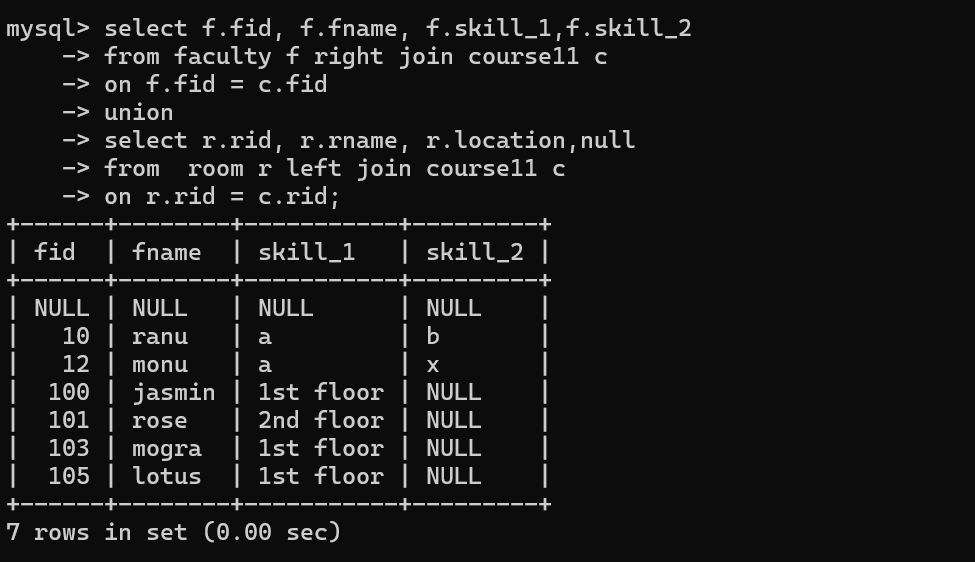
on f.fid = c.fid

union

select r.rid, r.rname, r.location,null

from room r left join course11 c

on r.rid = c.rid;

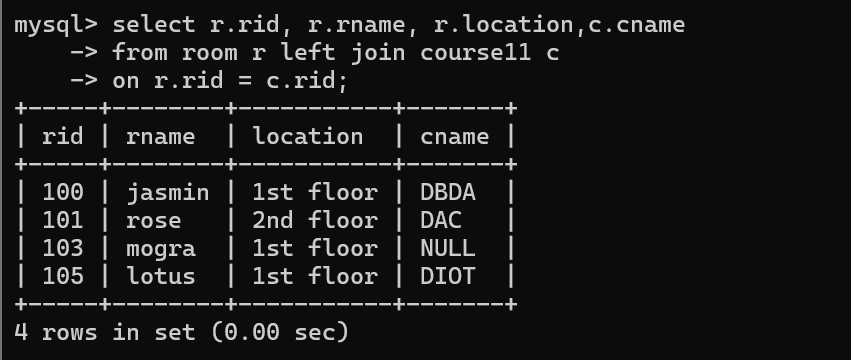


1. **list all rooms which are allocated or not allocated to any courses**

select r.rid, r.rname, r.location,c.cname

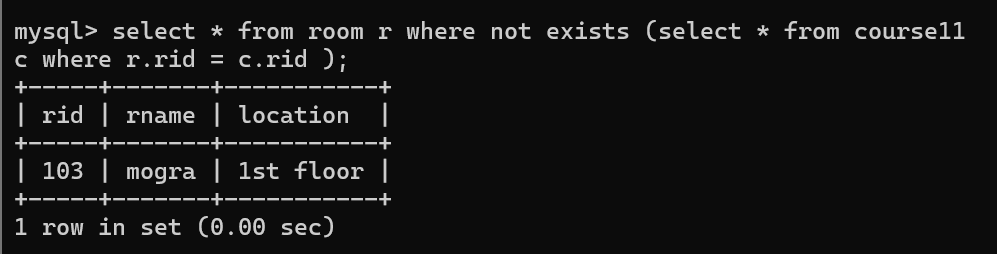
-> from room r left join course11 c

-> on r.rid = c.rid;



1. **list all rooms which are not allocated to any courses**

select \* from room r where not exists (select \* from course11 c where r.rid = c.rid );



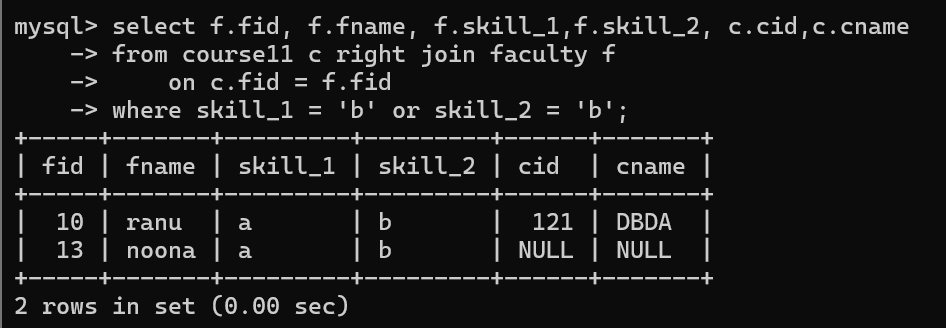
1. **display courses and faculty assigned to those courses whose special skill is database**

select f.fid, f.fname, f.skill\_1,f.skill\_2, c.cid,c.cname

from course11 c right join faculty f

on c.fid = f.fid

where skill\_1 = 'b' or skill\_2 = 'b';



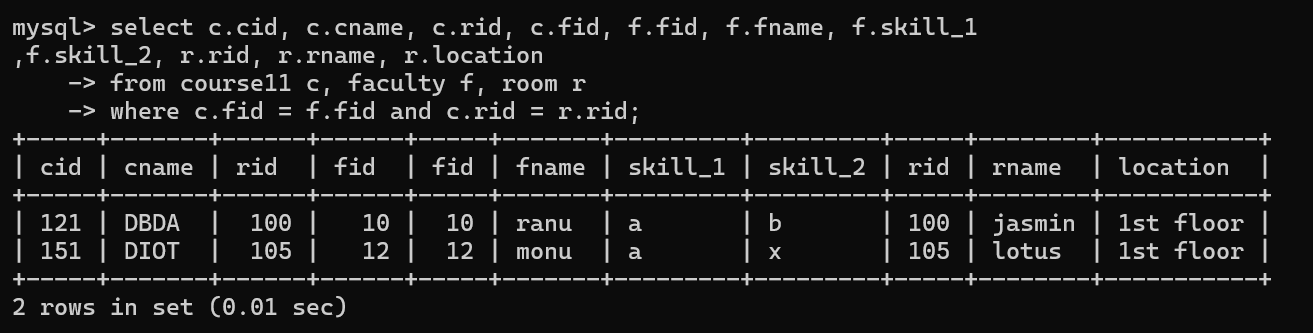
1. **display time table --- it should contain course details , faculty and room details**

select c.cid, c.cname, c.rid, c.fid, f.fid, f.fname, f.skill\_1

,f.skill\_2, r.rid, r.rname, r.location

from course11 c, faculty f, room r

where c.fid = f.fid and c.rid = r.rid;



**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**

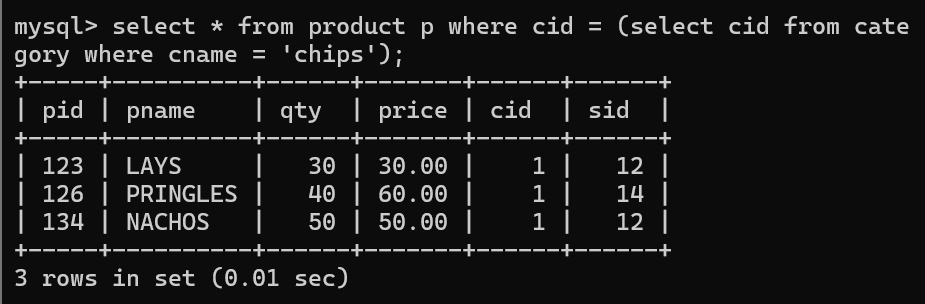
**14 Arnav Amaravati**

**category ---- cname unique and not null**

**cid cname description**

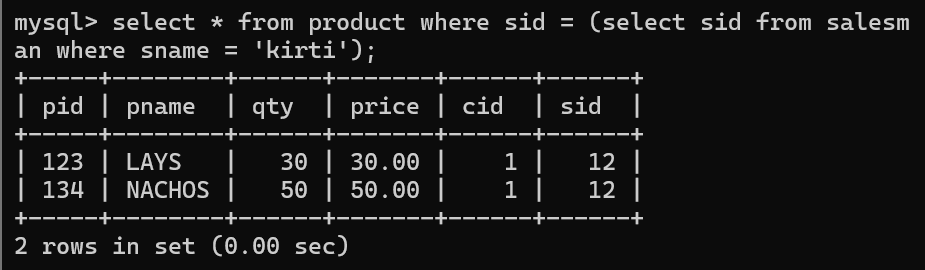
1. **chips very crunchy**
2. **chocolate very chocolaty**
3. **snacks yummy**
4. **cold drinks thanda thanda cool cool**
5. **List all products with category chips**

select \* from product p where cid = (select cid from category where cname = 'chips');



1. **display all products sold by kirti**

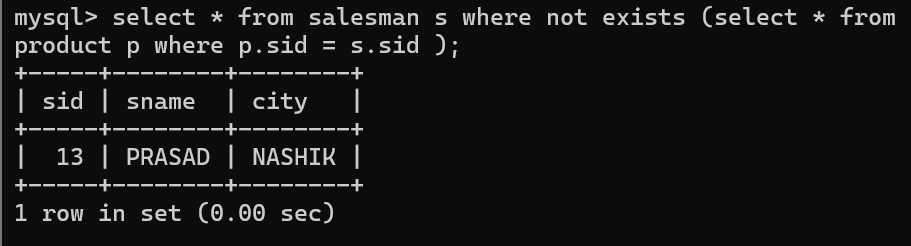
select \* from product where sid = (select sid from salesman where sname = 'kirti');



1. **display all salesman who do not sold any product**

select \* from salesman s where not exists (select \* from

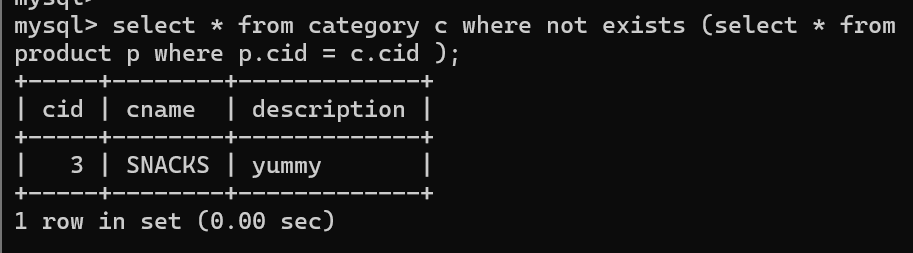
product p where p.sid = s.sid );



1. **display all category for which no product is there**

select \* from category c where not exists (select \* from

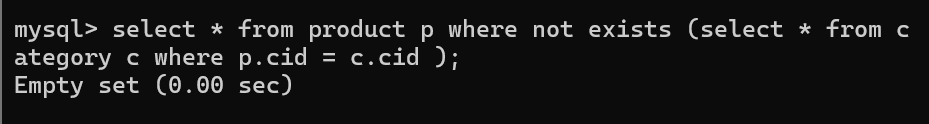
product p where p.cid = c.cid );



1. **display all products with no category assigned**

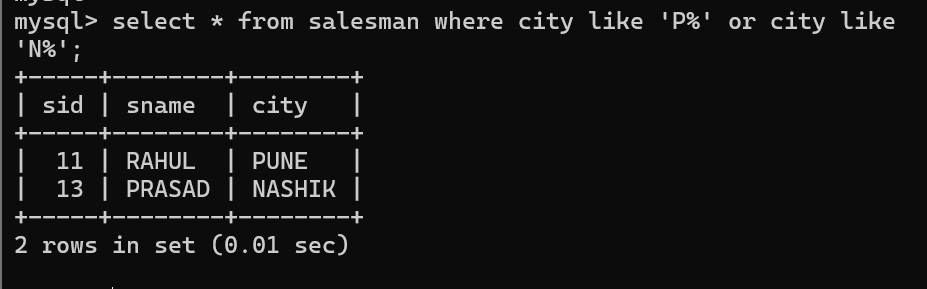
select \* from product p where not exists (select \* from c

ategory c where p.cid = c.cid );



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

select \* from salesman where city like 'P%' or city like 'N%';



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

