practice DQL statement

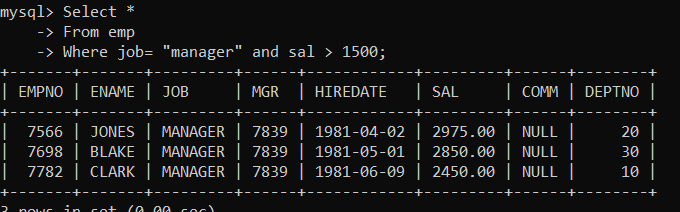
Write SQL statement for the following

1. To find all managers with salary >1500

Ans. Select \*

From emp

Where job= “manager“ and sal > 1500;

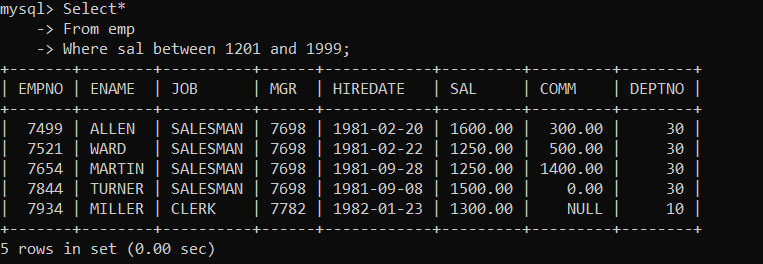


2. list all employees with sal >1200 and < 2000

Select\*

From emp

Where sal between 1201 and 1999;

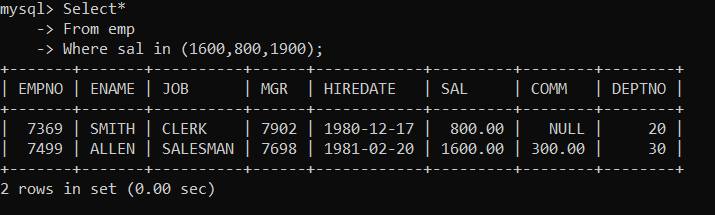


2. list all employees with sal is 1600 or sal is 800 or sal is 1900

Select\*

From emp

Where sal in (1600,800,1900);

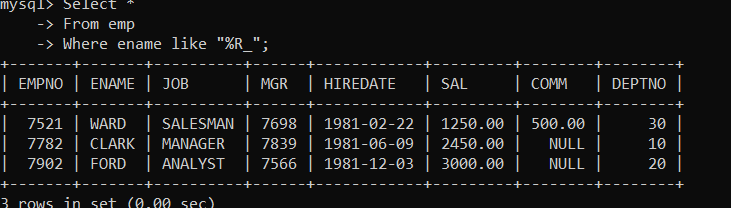


4. list all employees with R at second last position in name

Select \*

From emp

Where ename like “%R\_”;

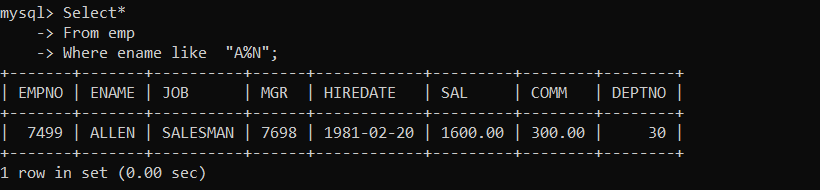


5. List all employees with name starts with A and ends with N

Select\*

From emp

Where ename like “A%N”;



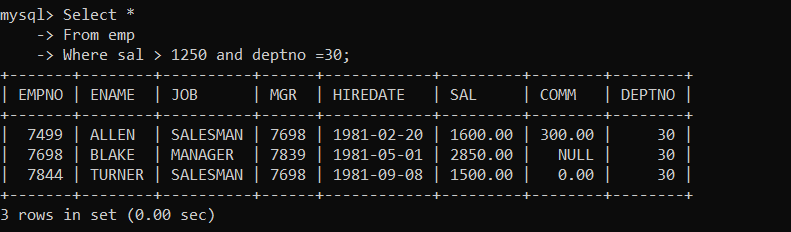
Q2. Solve following

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

Select \*

From emp

Where sal > 1250 and deptno =30;

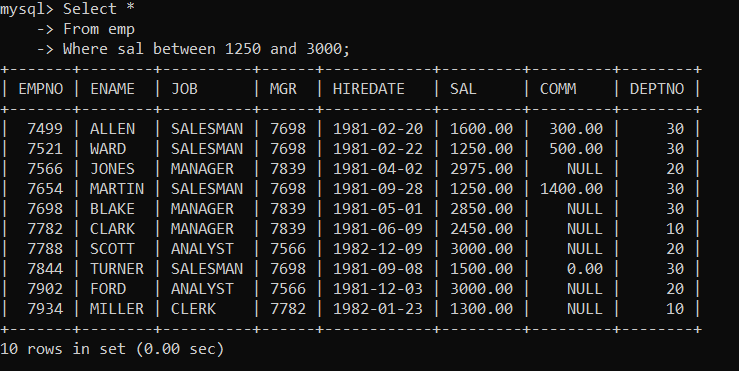


2. list all employees with salary >=1250 and <= 3000

Select \*

From emp

Where sal between 1250 and 3000;

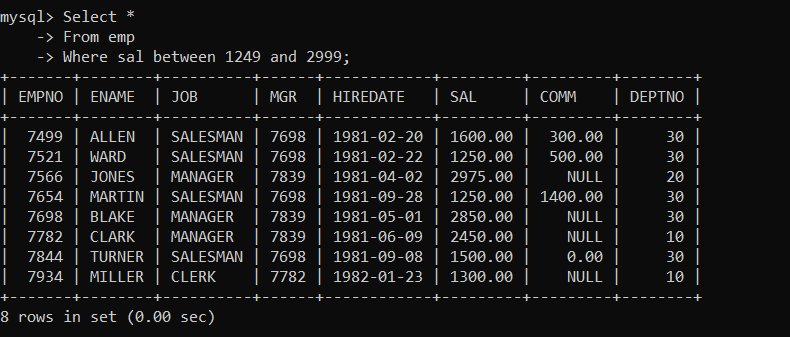


3. list all employees with salary >1250 and < 3000

Select \*

From emp

Where sal between 1249 and 2999;

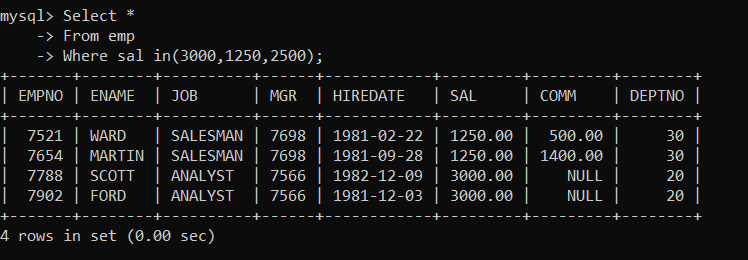


4. list all employees with salary either equal to 3000 or 1250 or 2500

Select \*

From emp

Where sal in(3000,1250,2500);

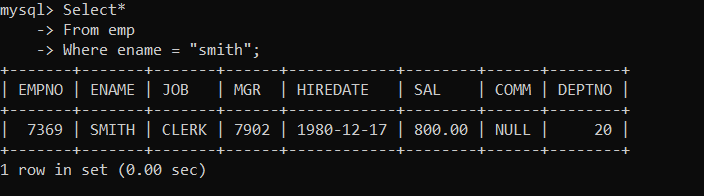


5. list all employee with name=SMITH

Select\*

From emp

Where ename = “smith”;

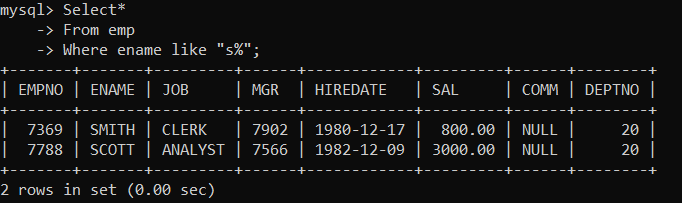


6. list all employees with name starting with S

Select\*

From emp

Where ename like “s%”;



7. list all employees with name ending with S

Select\*

From emp

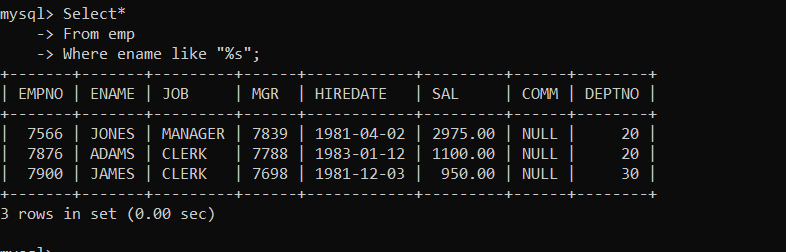
Where ename like “%s”;

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

Select\*

From emp

Where ename like “\_i%”;

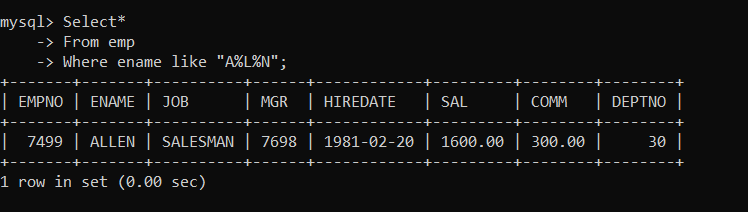


9. list all employees with name starts with A ends with N and somewhere in between L is there

Select\*

From emp

Where ename like “A%L%N”;

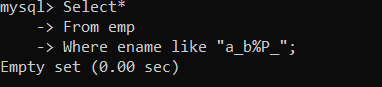


10. list all employees with name starts with A and B at 3 rd position and P at second last position.

Select\*

From emp

Where ename like “a\_b%P\_”;

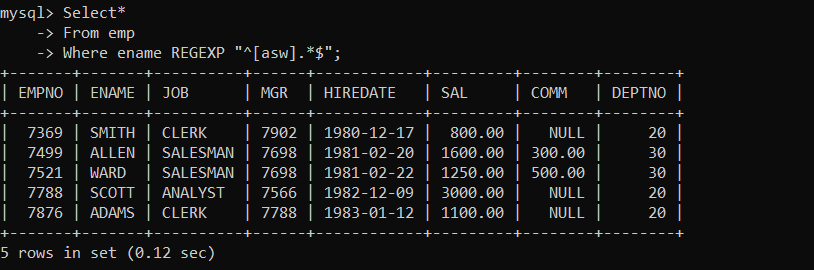


11. List all employees with name starts with either A or starts with S or starts with W

Select\*

From emp

Where ename REGEXP “^[asw].\*$”;



practice Aggregate functions

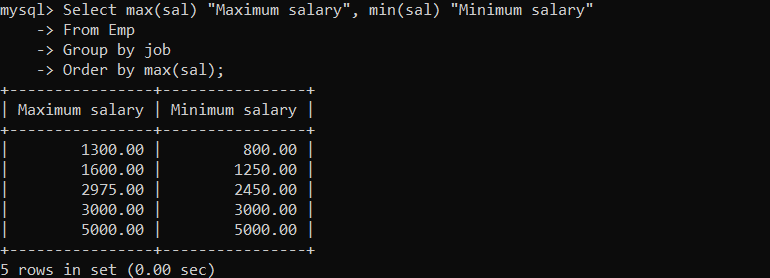
12. find max sal and min sal for each job

Select max(sal) “Maximum salary”, min(sal) “Minimum salary”

From Emp

Group by job

Order by max(sal);

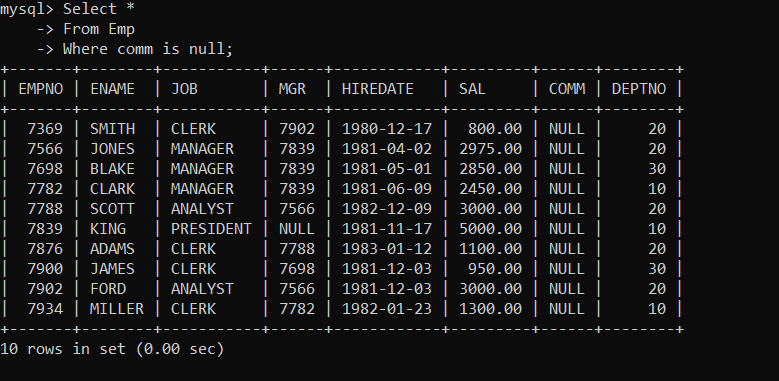


13. find how many employess have not received commission

Select \*

From Emp

Where comm is null;

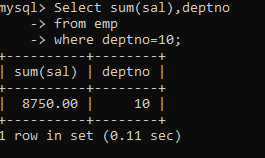


14. find sum of sal of all employees working in dept no 10

Select sum(sal),deptno

From emp

where deptno=10;

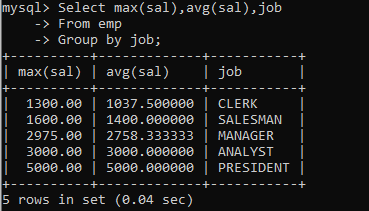


15. find maximum salary,average sal for each job in every department

Select max(sal),avg(sal),job

From emp

Group by job;



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

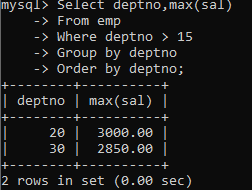
Select deptno,max(sal)

From emp

Where deptno > 15

Group by deptno

Order by deptno;



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

Ans. Select sum(sal) Total,deptno

From emp

Having sum(sal) > 3000;

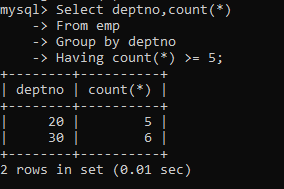
18. list all department which has minimum 5 employees

Ans. Select deptno,count(\*)

From emp

Group by deptno

Having count(\*) >= 5;



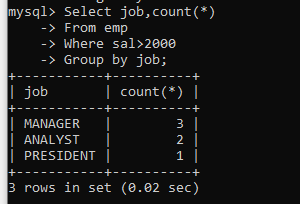
19. count how many employees earn salary more than 2000 in each job

Select job,count(\*)

From emp

Where sal>2000

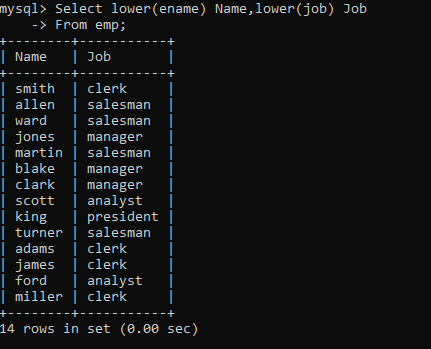
Group by job;



20. list all enames and jobs in small case letter

Select lower(ename) Name,lower(job) Job

From emp;

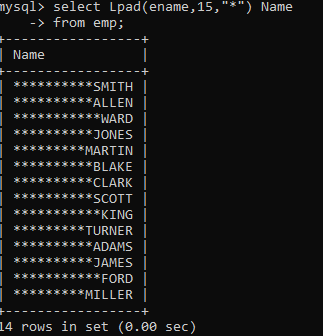


21. list all names and jobs so that the length of name should be 15 if it is smaller then add spaces

to left

select Lpad(ename,15,”\*”) Name

from emp;



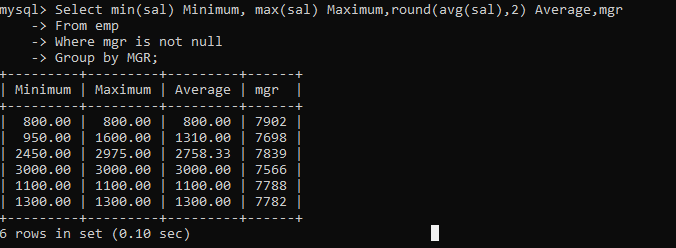
22. display min sal,max sal, average sal for all employees working under same manager

Select min(sal) Minimum, max(sal) Maximum,round(avg(sal),2) Average,mgr

From emp

Where mgr is not null

Group by MGR;



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

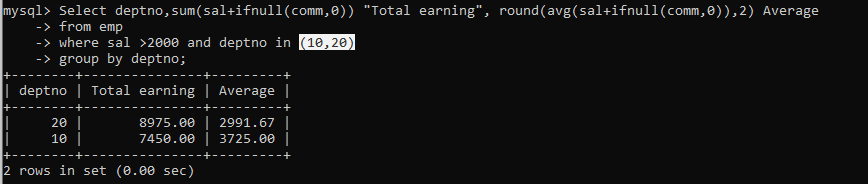
2000 and work in either dept no 10 or 20.

Select deptno,sum(sal+ifnull(comm,0)) “Total earning”, round(avg(sal+ifnull(comm,0)),2) Average

From emp

Where deptno in (10,20)

Group by deptno;

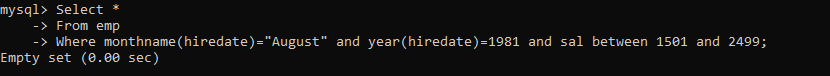


24. list all employees who joined in Aug 1980 and salary is >1500 and < 2500

Select \*

From emp

Where monthname(hiredate)=”August” and year(hiredate)=1981 and sal between 1501 and 2499;

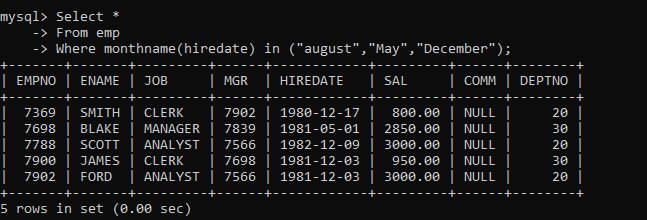


25. list all employees joined in either aug or may or dec

Select \*

From emp

Where monthname(hiredate) in (“august”,”May”,”December”);



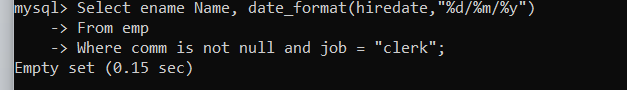
26. display name and hiredate in dd/mm/yy format for all employees whose job is clerk and they

earn some commission.

Select ename Name, date\_format(hiredate,”%d/%m/%y”)

From emp

Where comm is not null and job = ‘clerk’;

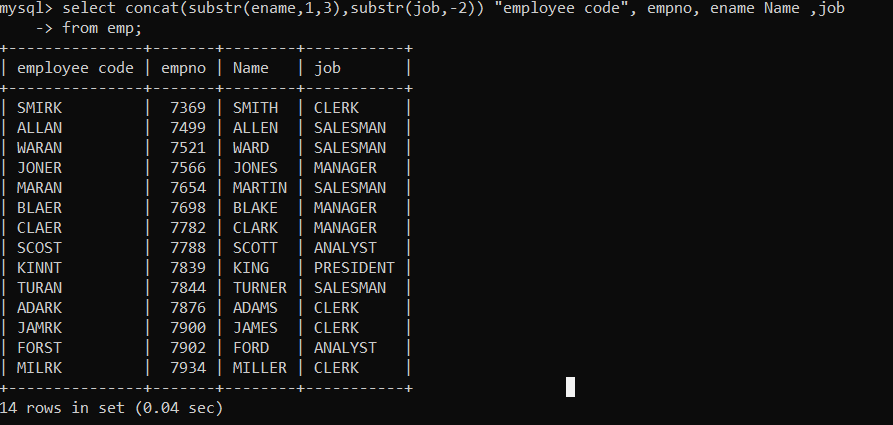


27. list empcode,empno,name and job for each employee. (note :empcode is 3 to 5 characters

from name and last 2 characters of job)

select concat(substr(ename,1,3),substr(job,-2)) “employee code”, empno, ename Name ,job

from emp;



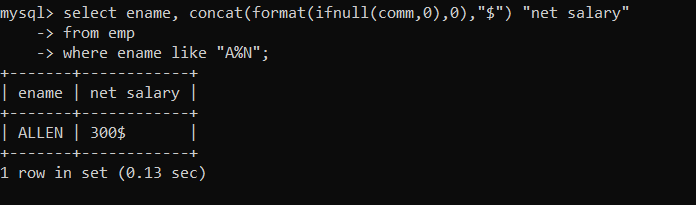
28. display thousand separator and $ symbol for commission if it is null then display it as 0 for all

employees whose name starts with A and ends with N

select ename, concat(format(ifnull(comm,0),0),”$”)

from emp

where ename like ‘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"

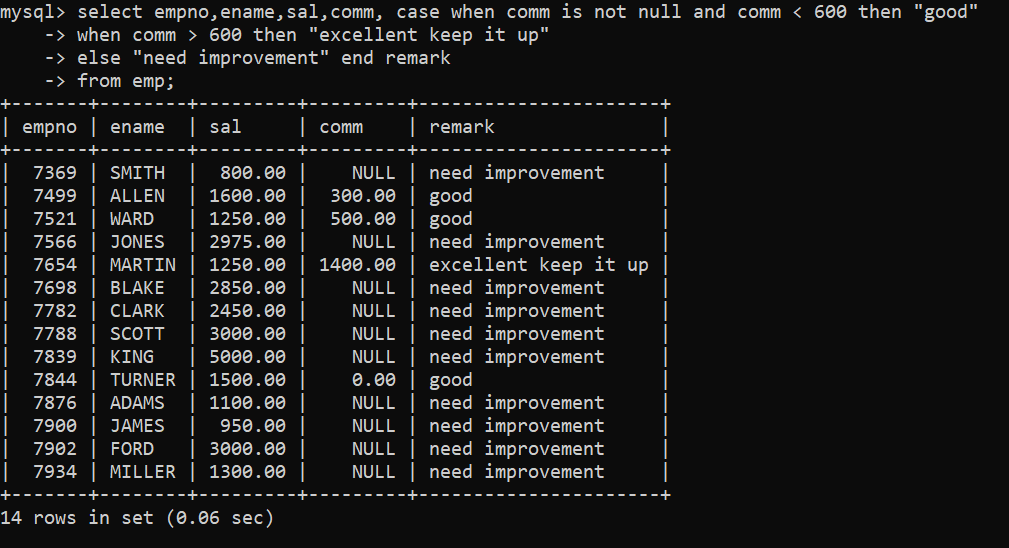
otherwise "Need improvement"

select empno,ename,sal,comm, case when comm is not null and comm < 600 then “good”

when comm > 600 then “excellent keep it up”

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

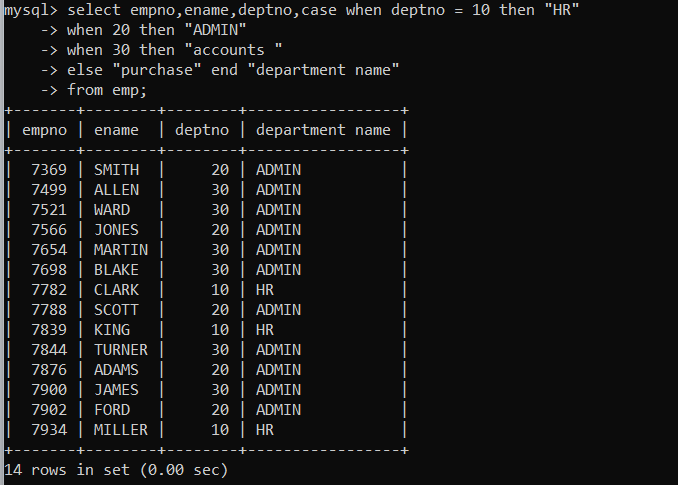
ans. select empno,ename,deptno,case when deptno = 10 then “HR”

when 20 then “ADMIN”

when 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 number primary key,

dname varchar2(20) not null unique,

dloc varchar2(20)

)

Ans. Create table mydept\_DBDA(

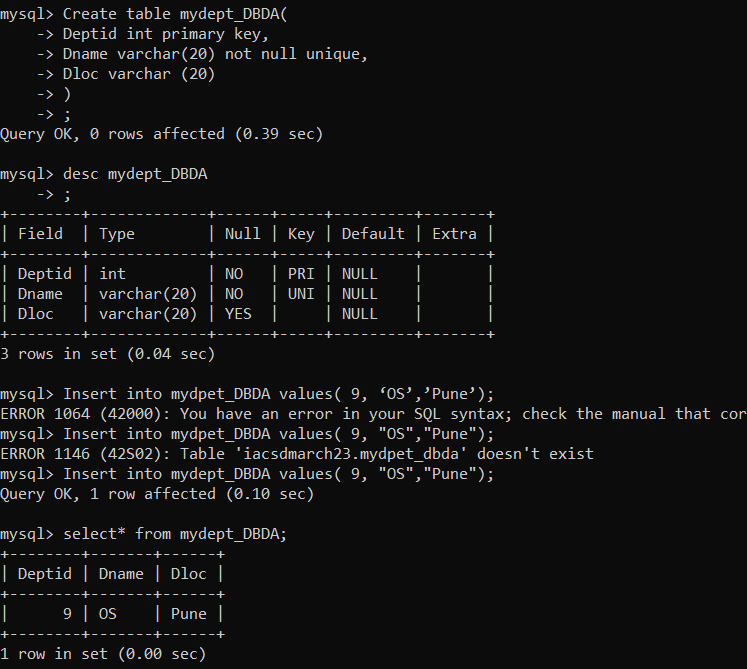
Deptid int primary key,

Dname varchar(20) not null unique,

Dloc varchar (20)

);

Insert into mydept\_DBDA values( 9, “OS”,”Pune”);



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

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

)

Create table employee(

Empno int,

Fname varchar(20) not null,

Mname varchar(15) ,

Lname varchar(15) not null,

Sal double(10,2),

Doj date,

Passportnumber varchar(20),

Deptno int,

constraint fk\_depno foreign key (deptno) references mydept\_dbda(deptid) on delete cascade on update cascade,

Constraint pk\_empno Primary key (empno),

Constraint un\_passportnum unique (passportnumber)

);

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)

)

Ans.

Create table student(  
sid int primary key,

Sname varchar(30) not null  
);

Create table course (

cid int primary key,

cname varchar(20) not null

);

Create table marks(

Sid int,

cid int,

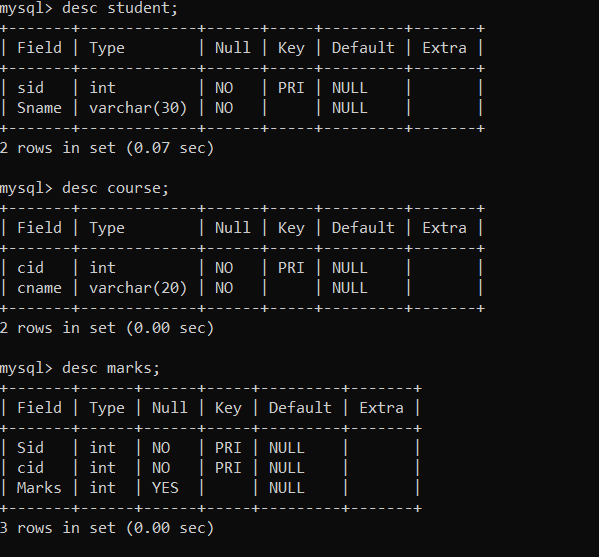
Marks int,

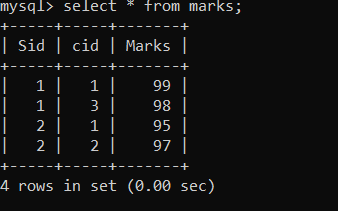
Constraint pk primary key(sid ,cid),

Constraint fk\_stuid Foreign key (sid) references student(sid) on delete cascade on update cascade,

Constraint fk\_courseid foreign key (cid) references course(cid) 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;

)

Create table empcopy as

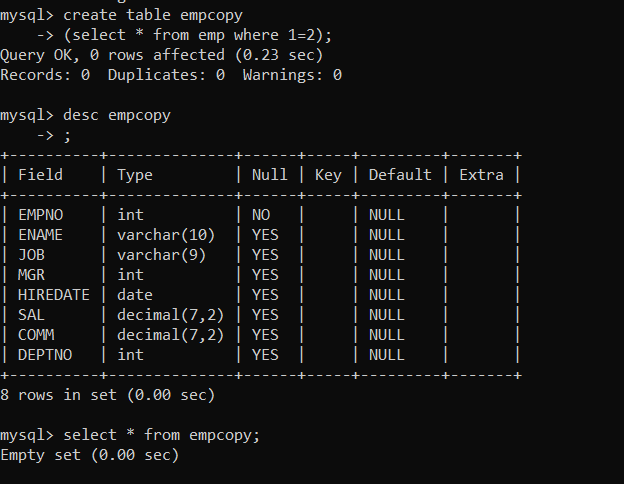
(

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

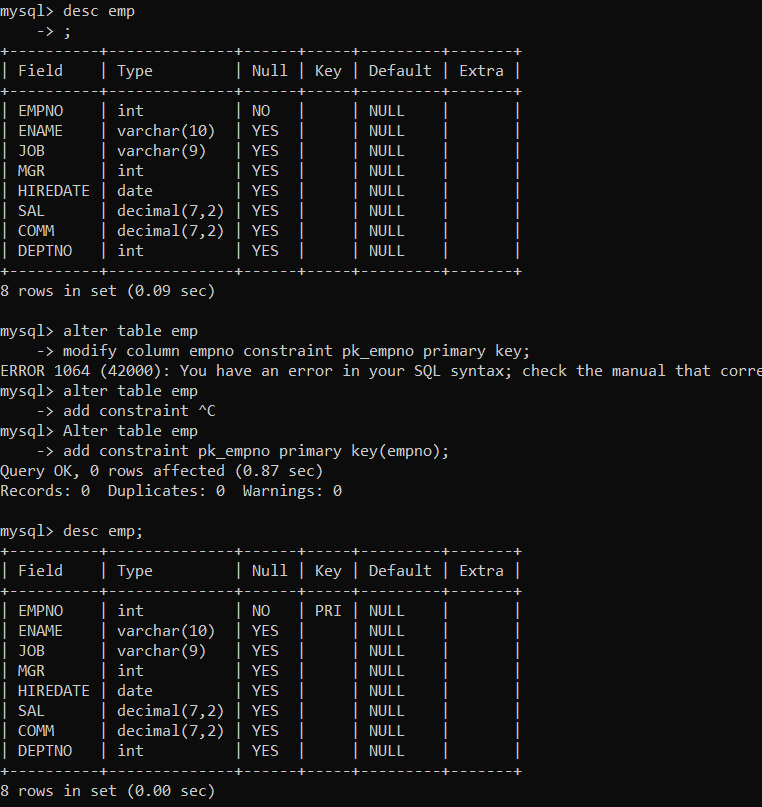
add foreign key constarint in emp

deptno --->> dept(deptno)

add new column in emp table netsal with constraint default 1000

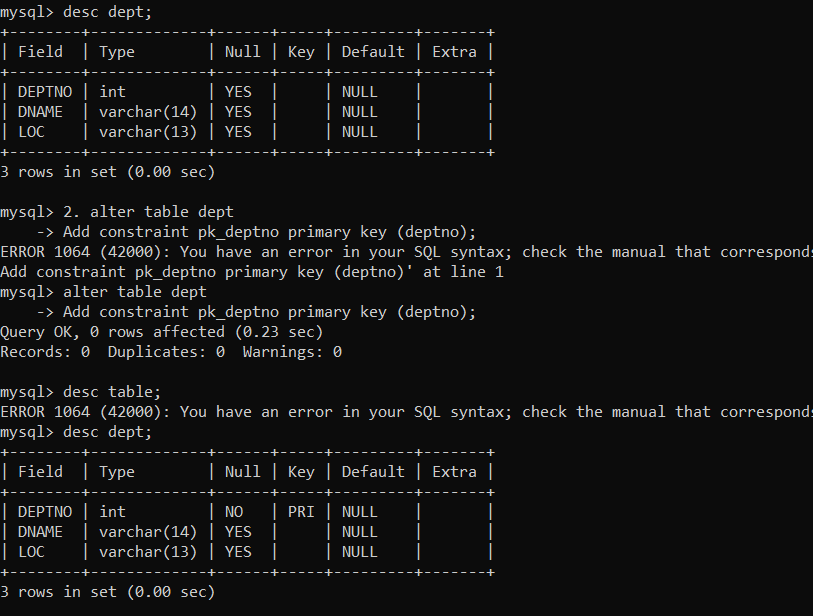
ans. 1.Alter table emp

add constraint pk\_empno primary key(empno);



2. alter table dept

Add constraint pk\_deptno primary key (deptno);

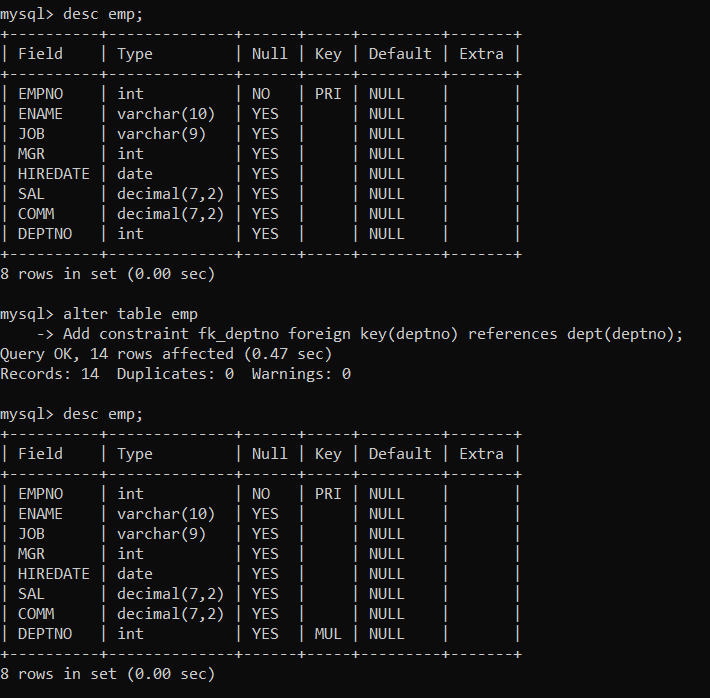


3. alter table salgrade

Add constraint pk\_grade primary key(grade);

4. alter table emp

Add constraint fk\_deptno foreign key(deptno) references dept(deptno);



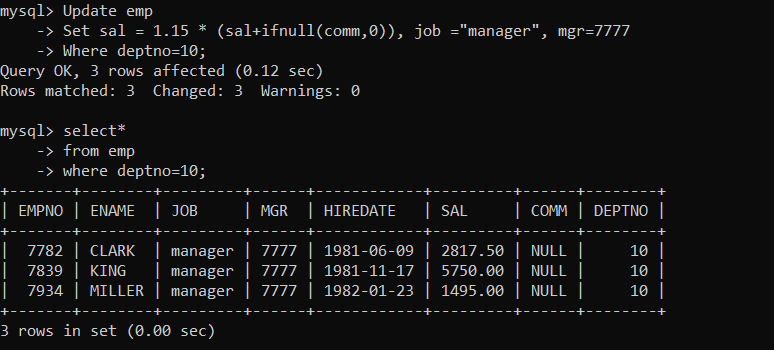
35. Update employee sal ---- increase sal of each employee by 15 % sal +comm, change the job to

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

Update emp

Set sal = 1.15 \* (sal+ifnull(comm,0)), job =”manager”, mgr=7777

Where deptno=10;

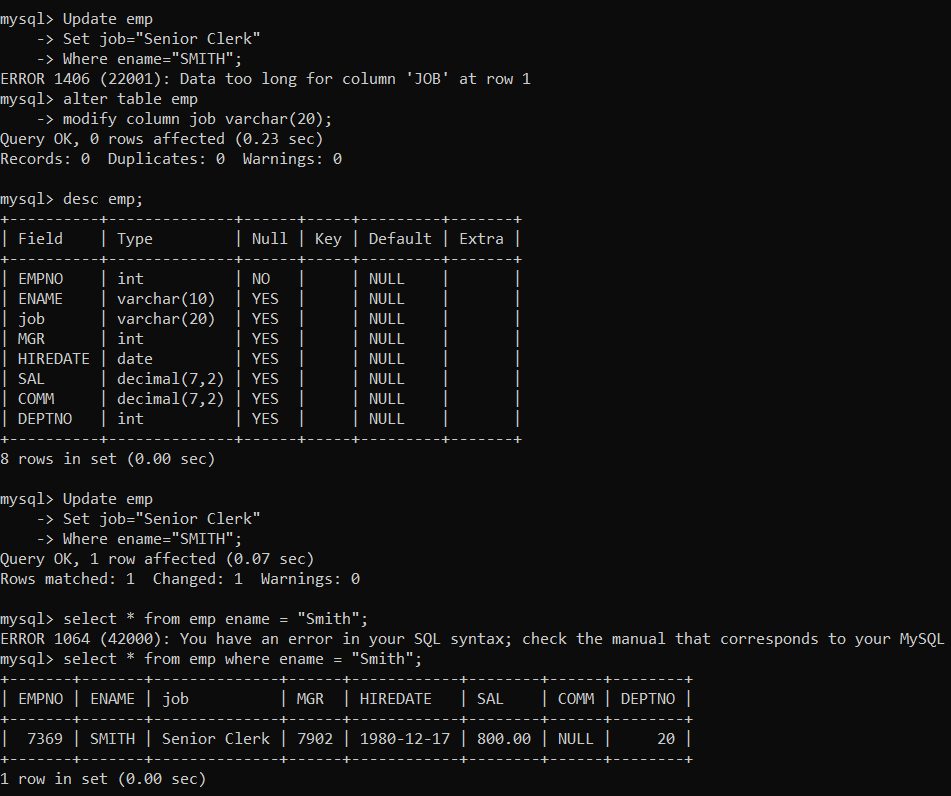


36. change job of smith to senior clerk

Update emp

Set job=”Senior Clerk”

Where ename=”SMITH”;

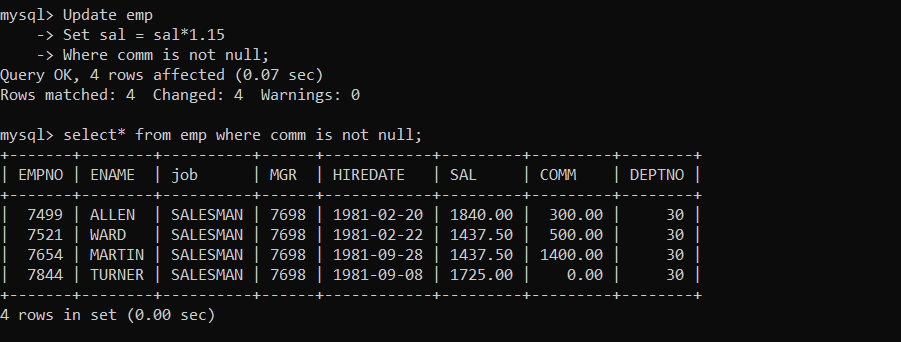


37. increase salary of all employees by 15% if they are earning some commission

Update emp

Set sal = sal\*1.15

Where comm is not null;

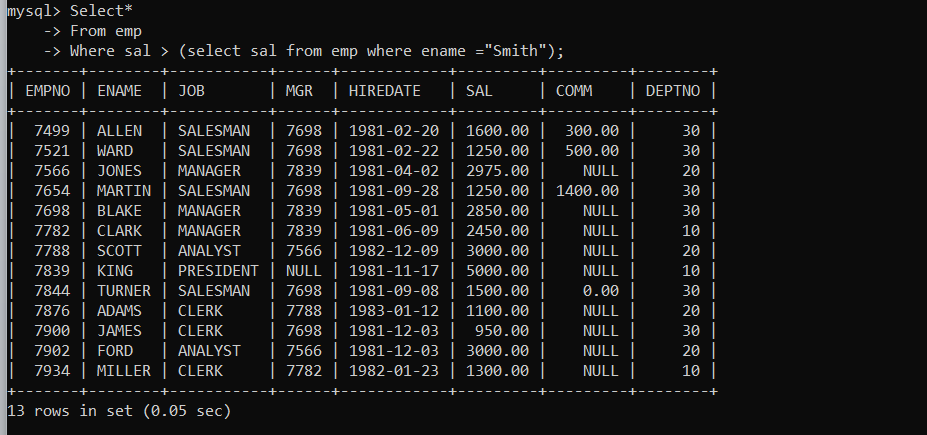


38. list all employees with sal>smith's sal

Select\*

From emp

Where sal > (select sal from emp where ename =”Smith”);

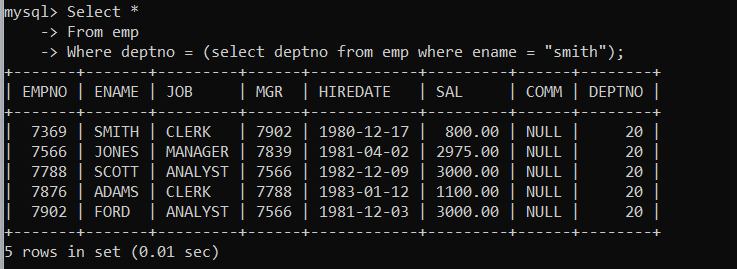


39. list all employees who are working in smith's department

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 < (select sal from emp where ename=”rajan”) and sal > (select sal from emp where ename=”revati”)

Order by sal;

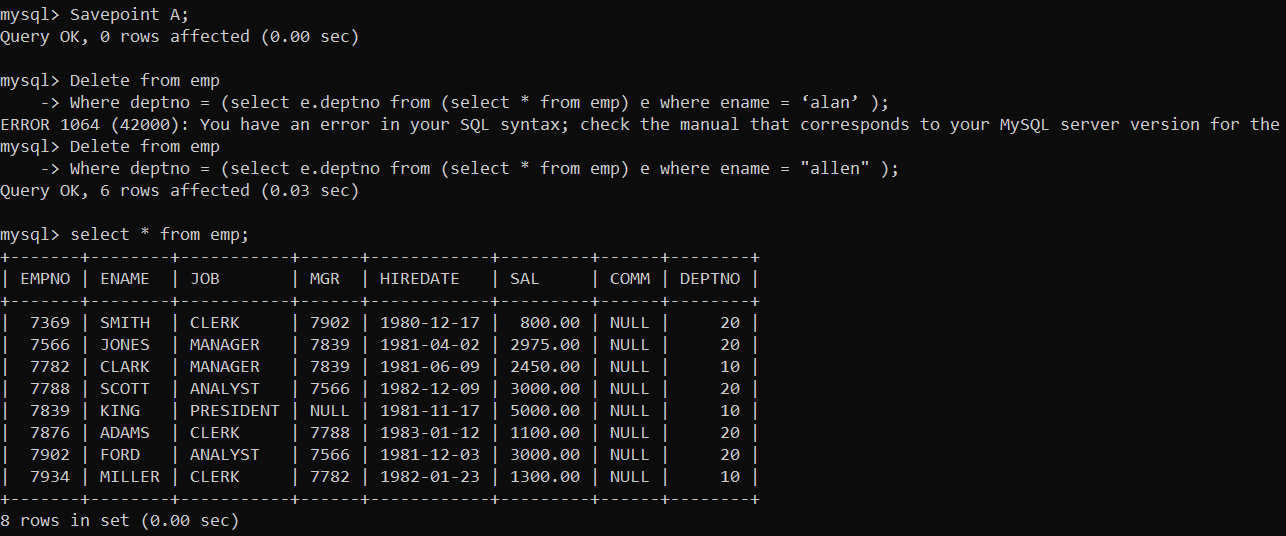
41. delete all employees working in alan's department

Set autocommit=0,

Savepoint A;

Delete from emp

Where deptno = (select e.deptno from (select \* from emp) e where ename = ‘alan’ );

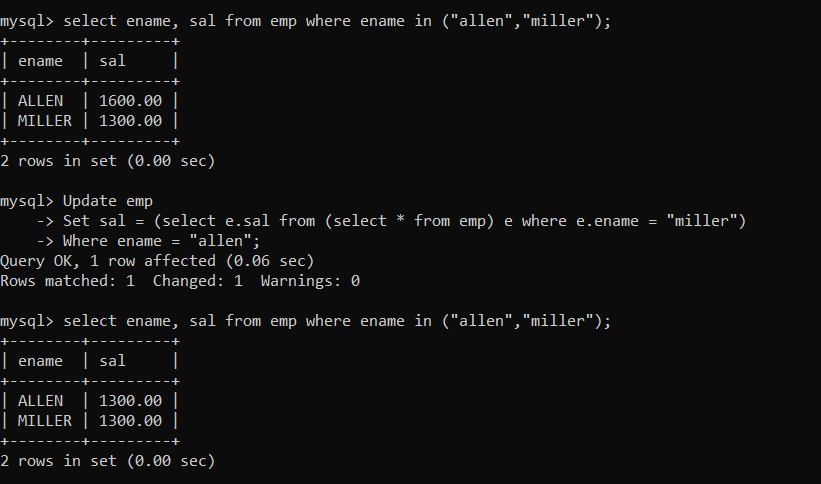


42. change salary of Alan to the salary of Miller.

Update emp

Set sal = (select e.sal from (select \* from emp) e where e.ename = “miller”)

Where ename = “allen”;



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

Update emp

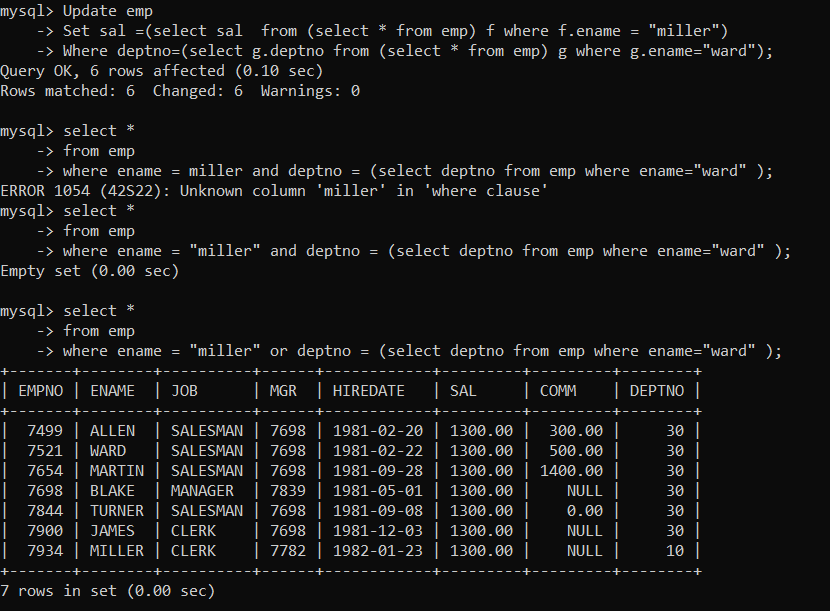
Set sal =(select sal from (select \* from emp) f where f.ename = “miller”)

Where deptno=(select g.deptno from (select \* from emp) g where g.ename=”ward”);

select \*

from emp

where ename = miller and deptno = (select deptno from emp where ename="ward" );

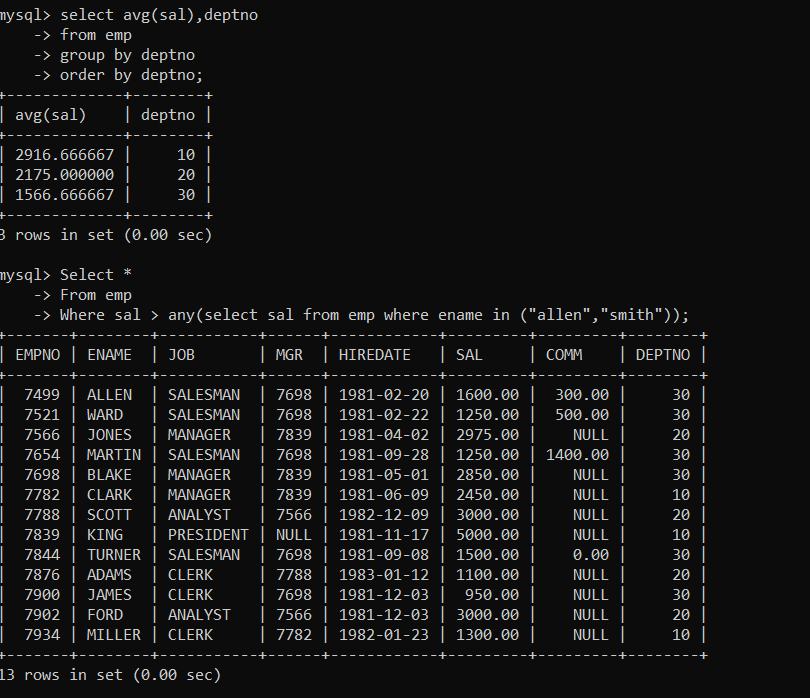


44. list all employees with salary > either Smith's salary or alan's sal

Select \*

From emp

Where sal > any(select sal from emp where ename in (“allen”,”smith”));

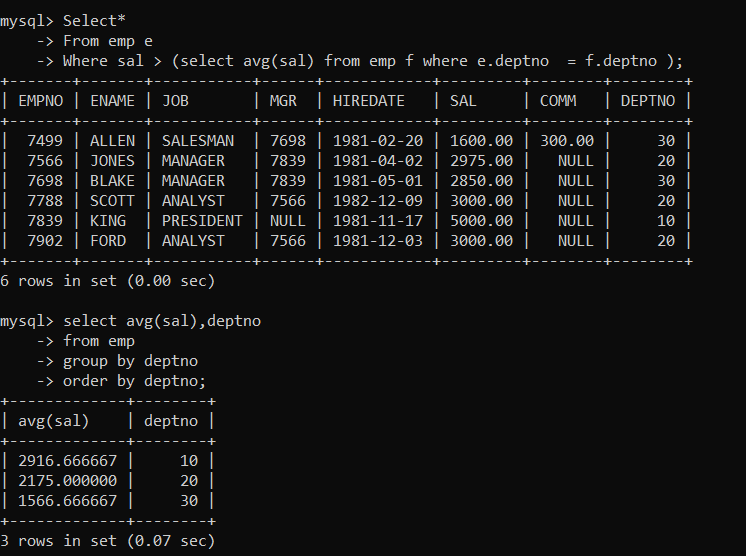


45. list all employees who earn more than average sal of dept 10

Select\*

From emp e

Where sal > (select avg(sal) from emp f where e.deptno = f.deptno );

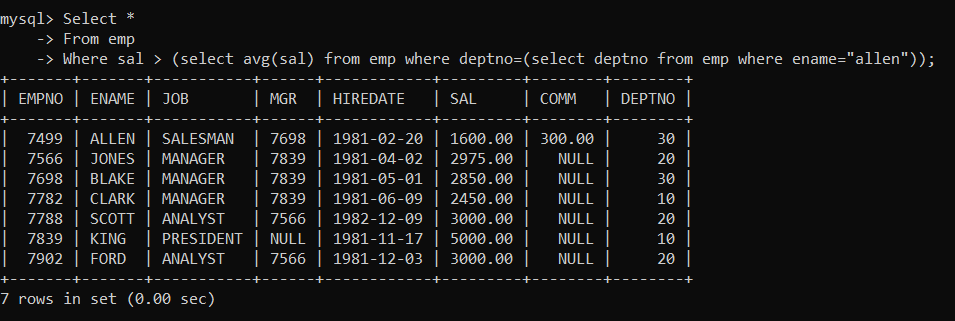


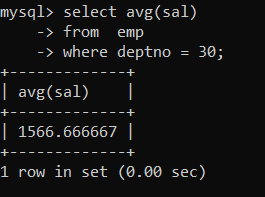
46. list all employees who earn more than average sal of Alan's department.

Select \*

From emp

Where sal > (select avg(sal) from emp where deptno=(select deptno from emp where ename=”allen”));





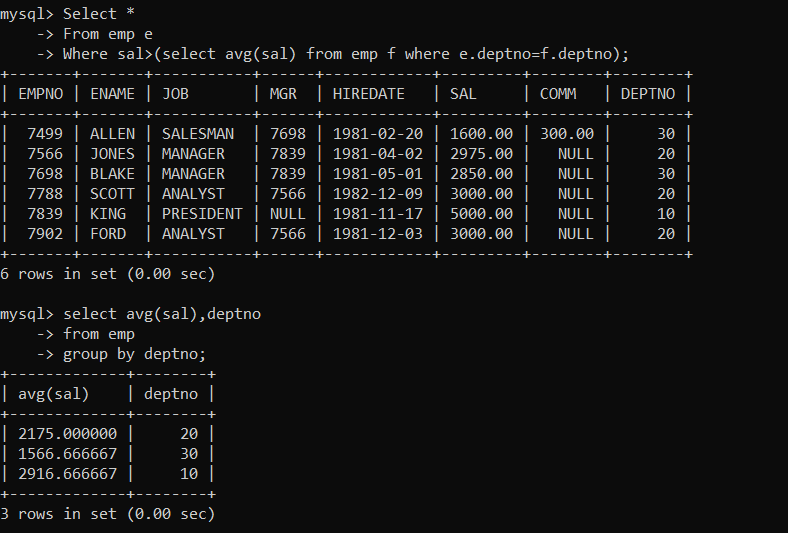
47. list all employees who are working in purchase department

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 f where e.deptno=f.deptno);

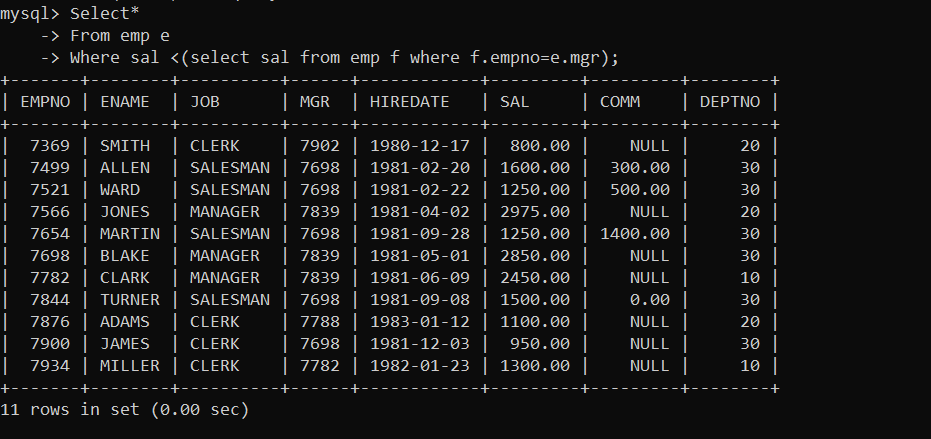


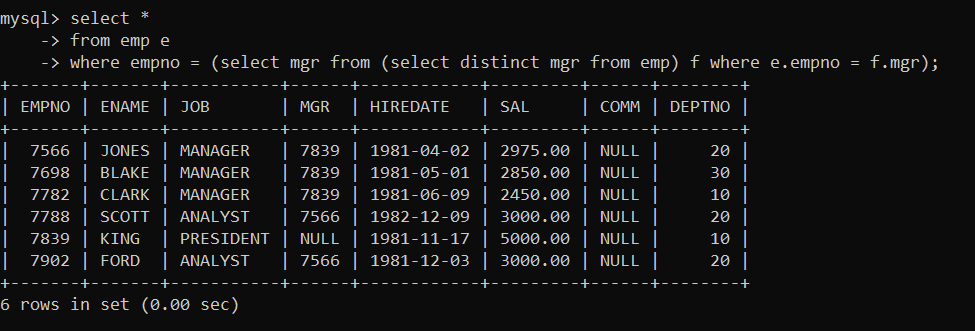
49. list all employees who earn sal < than their managers salary

Select\*

From emp e

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



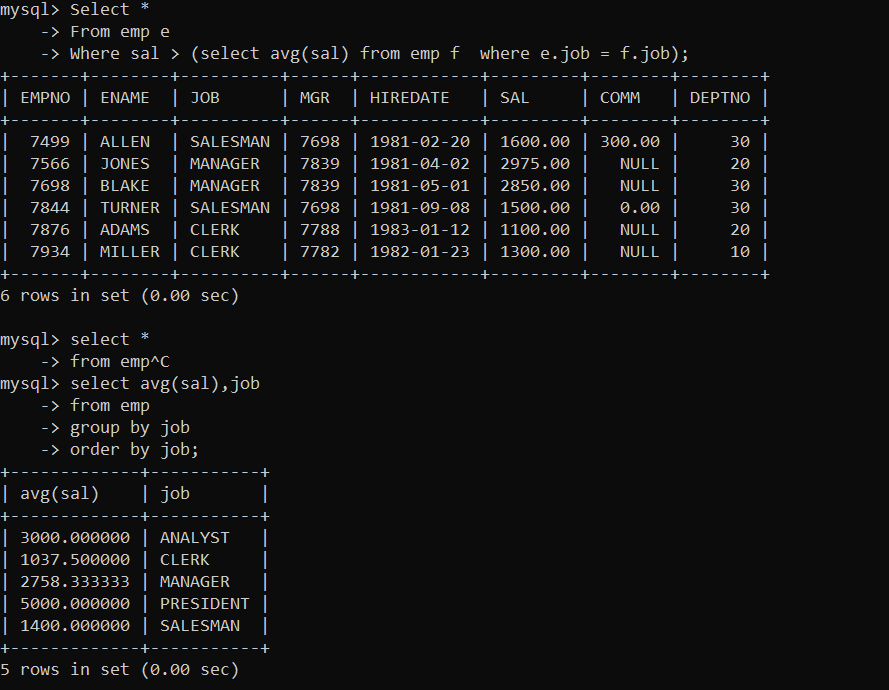


50. list all employees who are earning more than average salary of their job.

Select \*

From emp e

Where sal > (select avg(sal) from emp f where e.job = f.job):



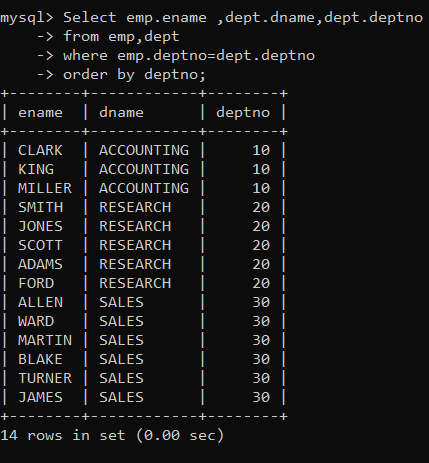
51. display employee name and department.

Select emp.ename ,dept.dname,dept.deptno

from emp,dept

where emp.deptno=dept.deptno

order by deptno;

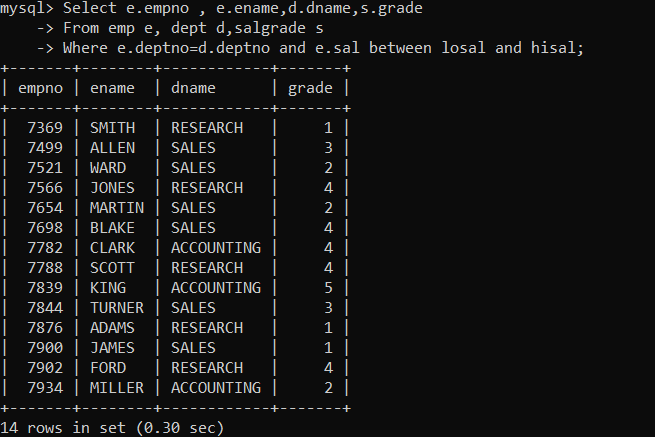


52. display empno,name,department name and grade (use emp,dept and salgrade table)

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

From emp e, dept d,salgrade s

Where e.deptno=d.deptno and e.sal between losal and hisal;

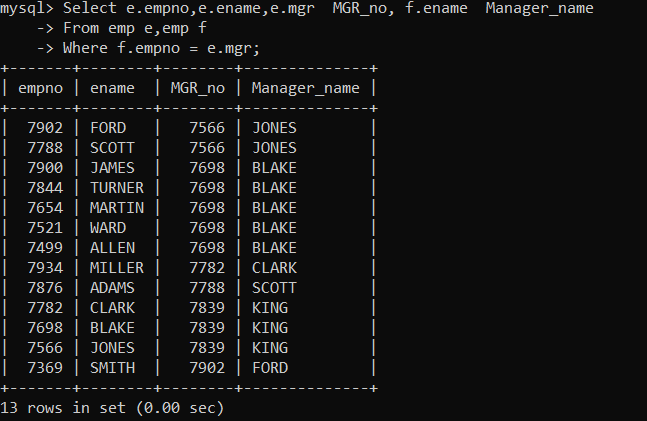


53. list all employees number,name, mgrno and manager name

Select e.empno,e.ename,e.mgr MGR\_no, f.ename Manager\_name

From emp e,emp f

Where f.empno = e.mgr;



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,cname,descritpion)

ans. create table salesman(

sid int,

sname varchar(30) not null,

address varchar(50) not null,

constraint pk\_sid primary key(sid)

);

Create table category(

cid int,

cname varchar(25) not null,

description text,

constraint pk\_cid primary key(cid)

);

Create table product(

Pid int primary key,

Pname varchar(20) not null,

Price double(7,2) default 100.00,

Qty smallint ,

Sid int,

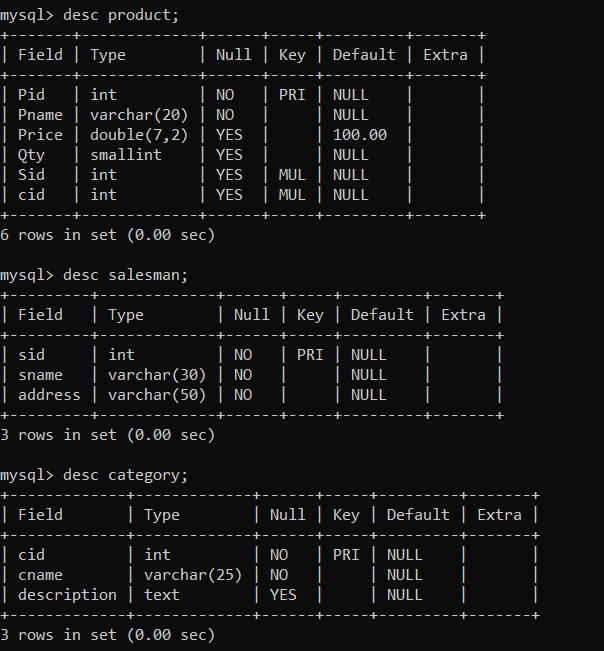
cid int,

constraint check\_price Check (price >0 and qty>0),

constraint fk\_sid foreign key (sid) references salesman(sid) on delete cascade on update cascade,

constraint fk\_cid foreign key (cid) references category(cid) on delete set null on update cascade

);

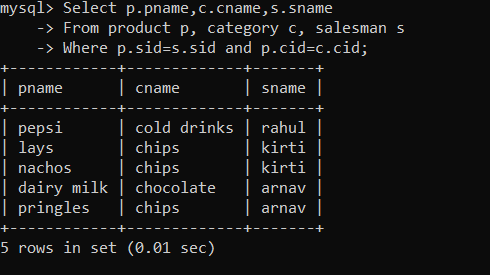


1. list all product name,their category name and name of a person, who sold that product.

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

From product p, category c, salesman s

Where p.sid=s.sid and p.cid=c.cid;

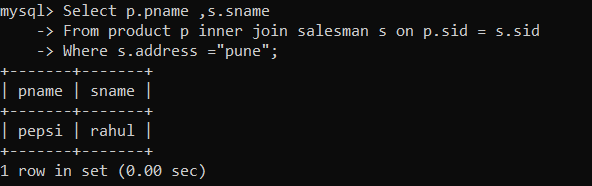


2. list all product name and salesman name for all salesman who stays in pune.

Select p.pname ,s.sname

From product p inner join salesman s on p.sid = s.sid

Where s.address =”pune”;



3. list all product name and category name

Select p.pname , c.cname

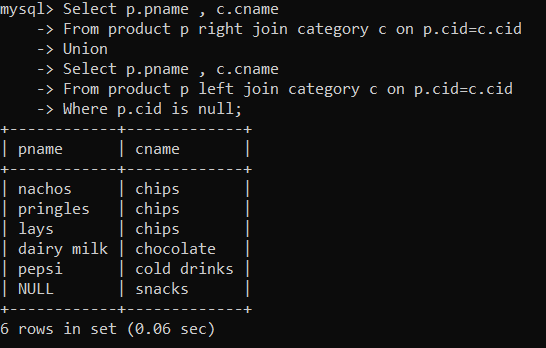
From product p right join category c on p.cid=c.cid

Union

Select p.pname , c.cname

From product p left join category c on p.cid=c.cid

Where p.cid is null;



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

Solution:

Create table course(

cid int primary key,

cname varchar(30) not null,

rid int,

fid int,

constraint rk\_rid foreign key (rid) references room(roomid) on delete set null on update cascade,

constraint fk\_fid foreign key (fid) references faculty(fid) on delete set null on update cascade

);

Create table faculty(

Fid int primary key,

Fname varchar(30) not null,

Sp\_skill1 varchar(30),

Sp\_skill2 varchar(30)

);

Create table room(

Roomid int primary key,

Rname varchar(30) not null,

Rloc varchar(30) not null

);

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”);

Insert into course values(121,”DBDA”,100 ,10);

Insert into course(cid,cname,rid) values(131,”DAC “,101);

Insert into course(cid,cname) values(141,”DTISS”);

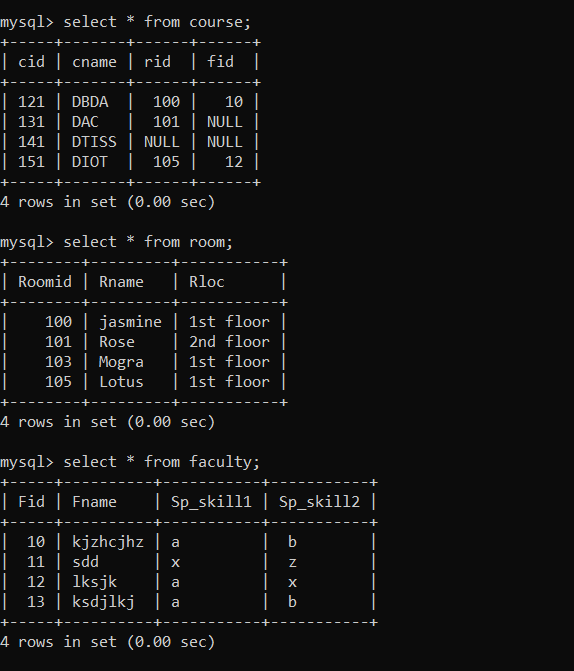
Insert into course values(151,”DIOT”,105,12);

Insert into room values(100,”jasmine”,”1st floor”);

Insert into room values(101,”Rose”,”2nd 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.

Select cname,rname

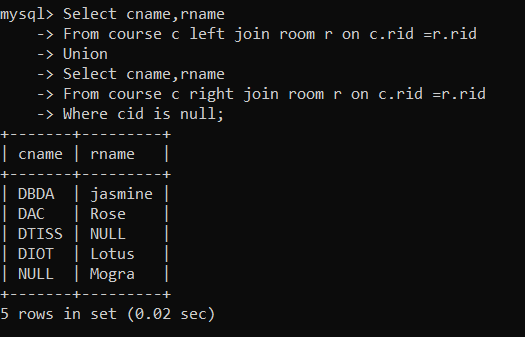
From course c left join room r on c.rid =r.rid

Union

Select cname,rname

From course c right join room r on c.rid =r.rid

Where cid is null;



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

allocated to any course.

Select fname ,cname,rname

From faculty f left join course c on f.fid = c.fid left join room r on r.rid =c.rid

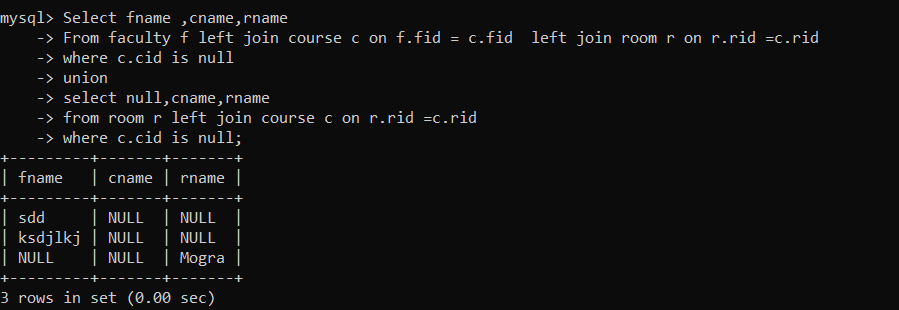
where c.cid is null

union

select null,cname,rname

from room r left join course c on r.rid =c.rid

where c.cid is null;

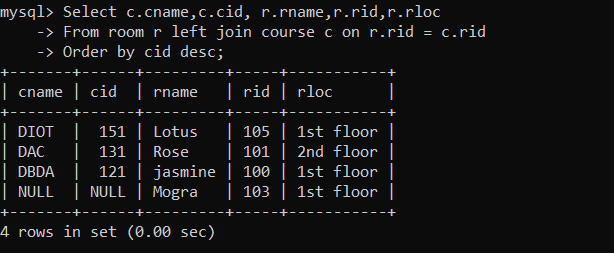


3. list all rooms which are allocated or not allocated to any courses.

Select c.cname,c.cid, r.rname,r.rid,r.rloc

From room r left join course c on r.rid = c.rid

Order by cid desc;

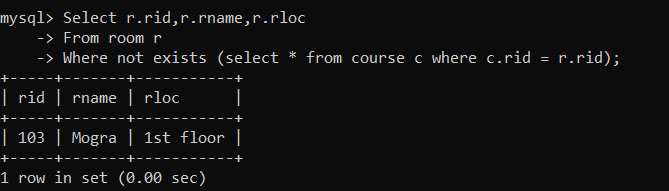


4. list all rooms which are not allocated to any courses.

Select r.rid,r.rname,r.rloc

From room r

Where not exists (select \* from course c where c.rid = r.rid);



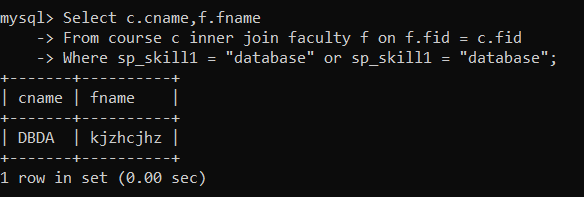
5. display courses and faculty assigned to those courses whose special skill is

Database.

Select c.cname,f.fname

From course c inner join faculty f on f.fid = c.fid

Where sp\_skill1 = “database” or sp\_skill1 = “database”;



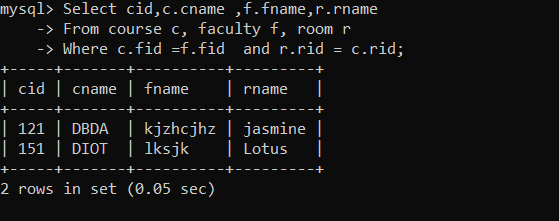
6. display time table --- it should contain course details , faculty and room

Details.

Select cid,c.cname ,f.fname,r.rname

From course c, faculty f, room r

Where c.fid =f.fid and r.rid = c.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’,60.00,40, 14,2

125,’pringles’,60.00,40,14,1

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

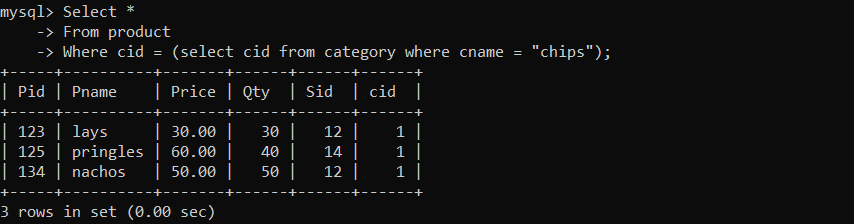
4 cold drinks thanda thanda cool cool

1. List all products with category chips.

Select \*

From product

Where cid = (select cid from category where cname = “chips”);

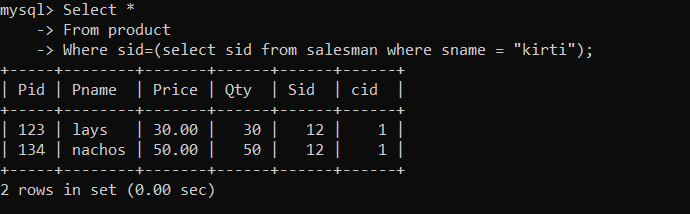


2. display all products sold by kirti.

Select \*

From product

Where sid=(select sid from salesman where sname = “kirti”);

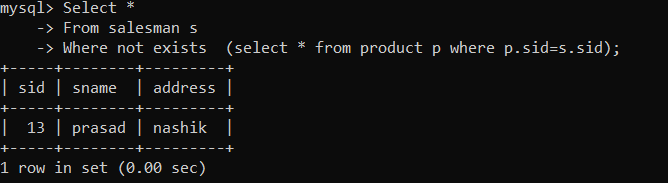


3. display all salesman who do not sold any product.

Select \*

From salesman s

Where not exists (select \* from product p where p.sid=s.sid);

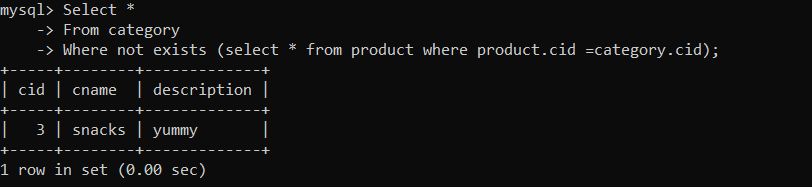


4. display all category for which no product is there.

Select \*

From category

Where not exists (select \* from product where product.cid =category.cid);

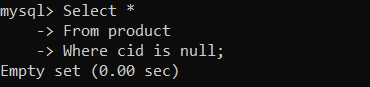


5. display all products with no category assigned.

Select \*

From product

Where cid is null;

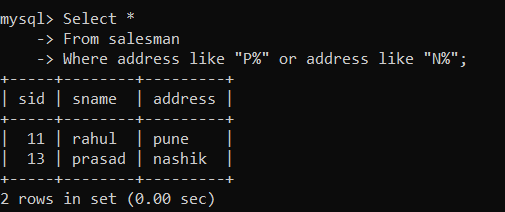


6. list all salesman who stays in city with name starts with P or N.

Select \*

From salesman

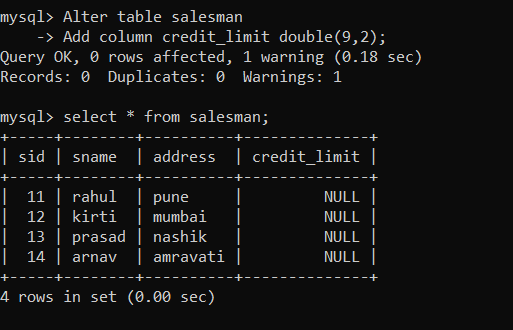
Where address like “P%” or address like “N%”;



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

Alter table salesman

Add column credit\_limit double(9,2);



------------------------------