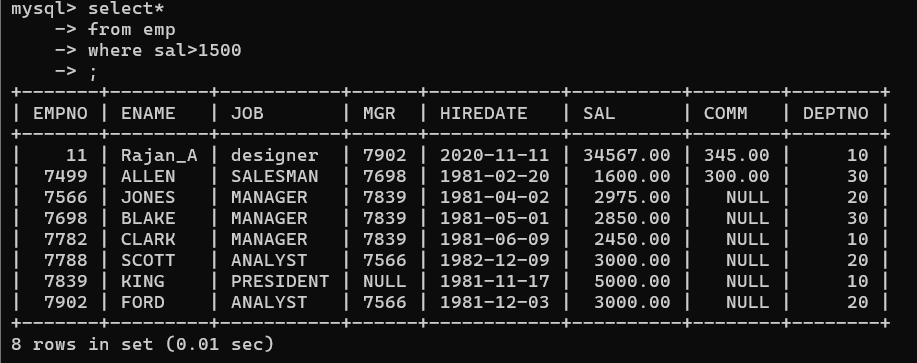
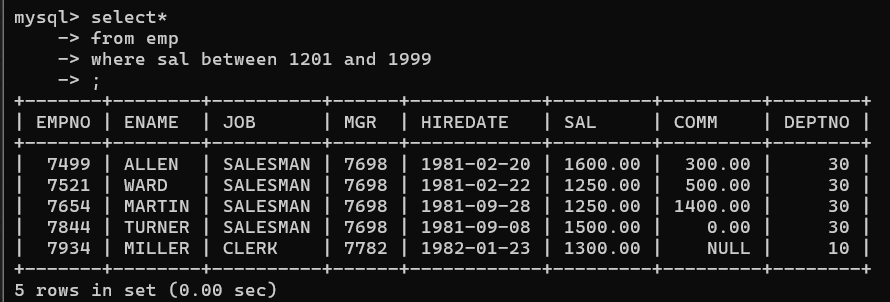
practice DQL statement

Write SQL statement for the following

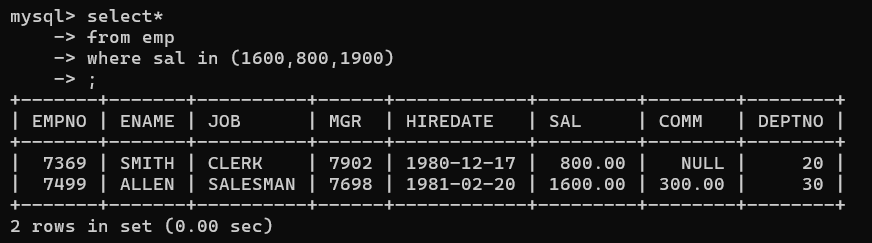
1. To find all managers with salary >1500



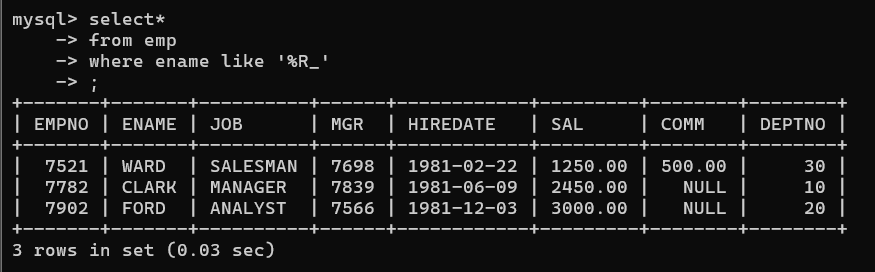
2. list all employees with sal >1200 and < 2000



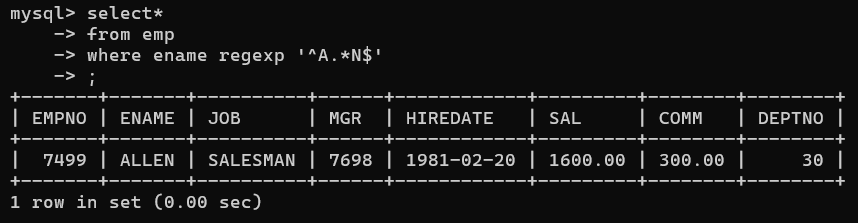
3. list all employees with sal is 1600 or sal is 800 or sal is 1900



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

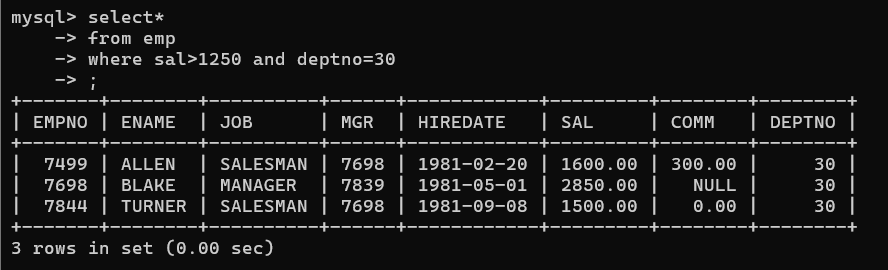


5. List all employees with name starts with A and ends with N

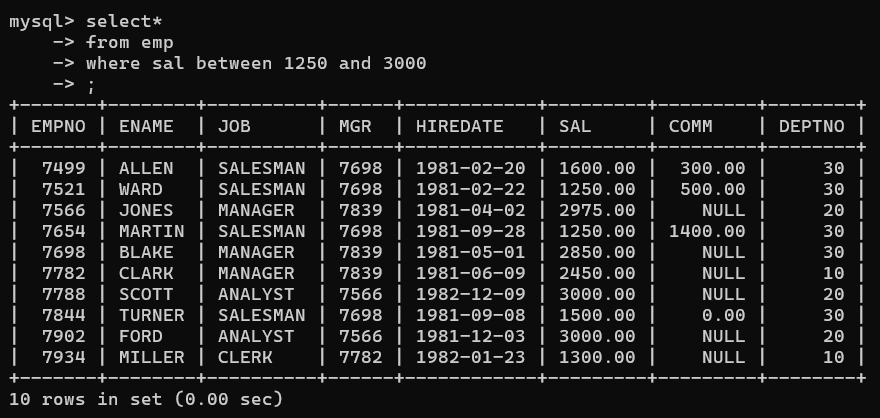


Q2. Solve following

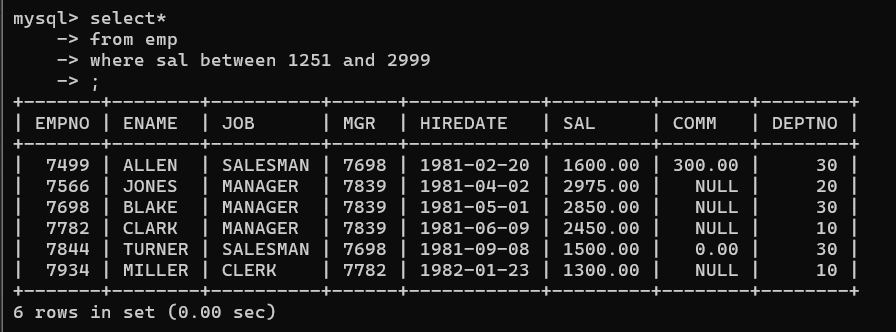
1. list all employees with salary > 1250 and dept no=30



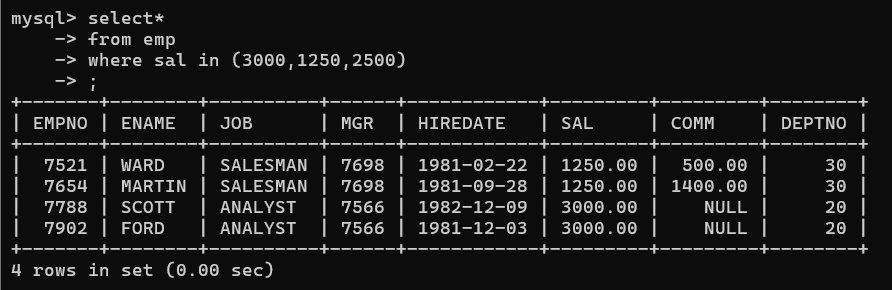
2.all employees with salary >=1250 and <= 3000



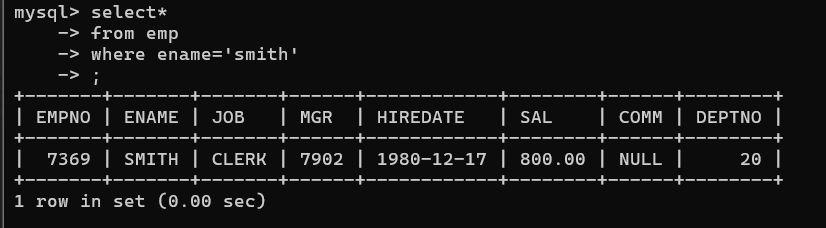
3.list all employees with salary >1250 and < 3000



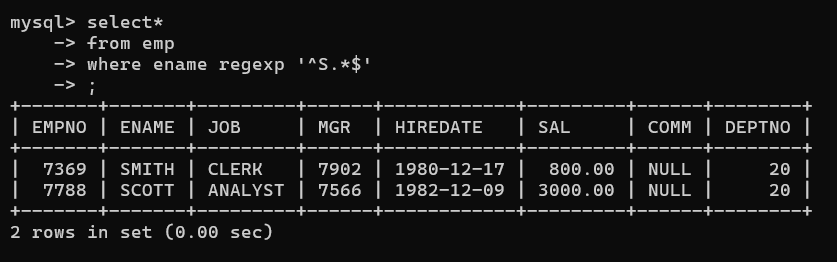
4. list all employees with salary either equal to 3000 or 1250 or 2500



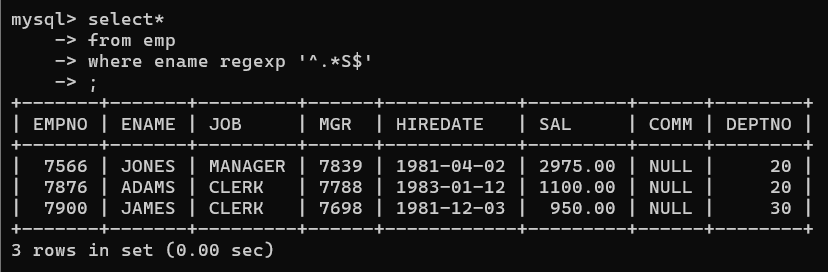
5. list all employee with name=SMITH



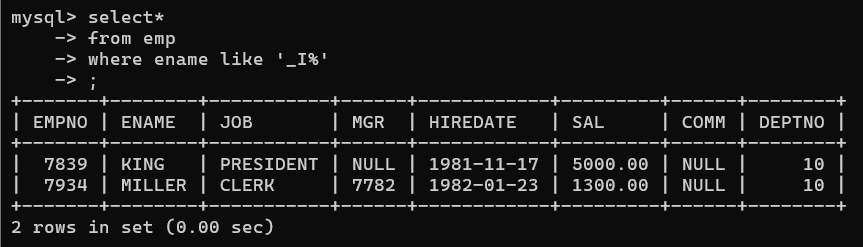
6. list all employees with name starting with S



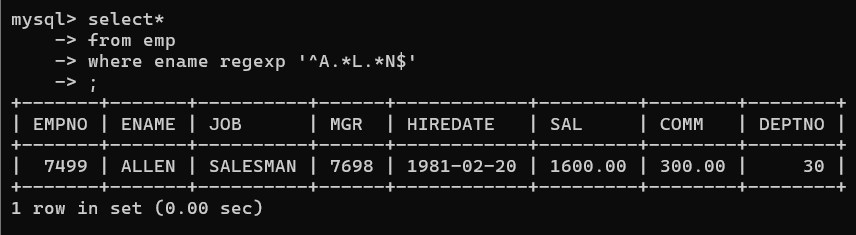
7. list all employees with name ending with S



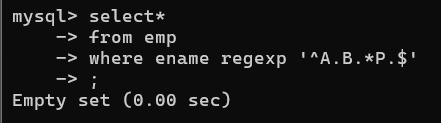
8. list all employees with name contains I at 2nd position



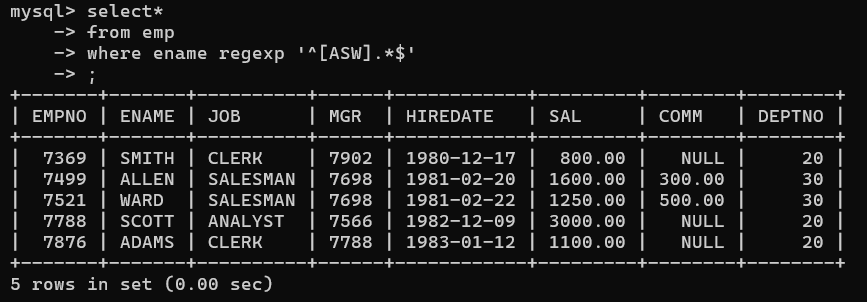
9. list all employees with name starts with A ends with N and somewhere in between L is there



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

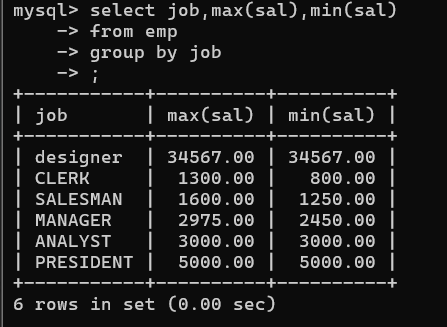


11. List all employees with name starts with either A or starts with S or starts with W

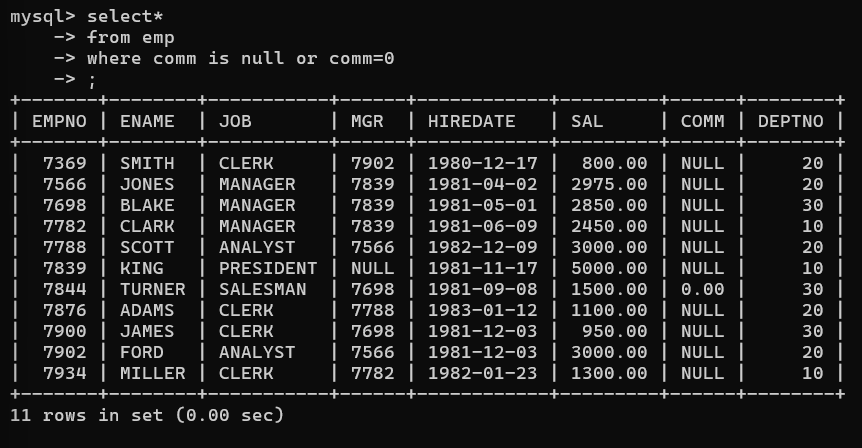


practice Aggregate functions

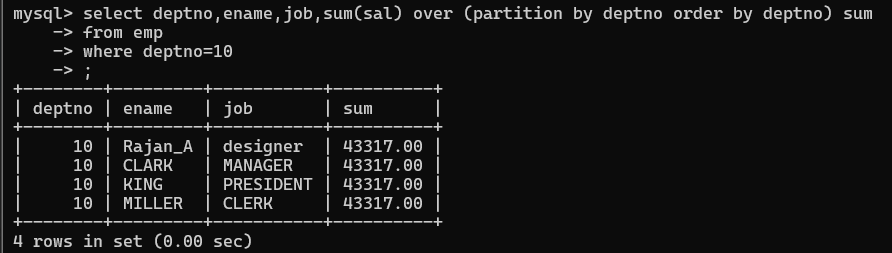
12. find max sal and min sal for each job



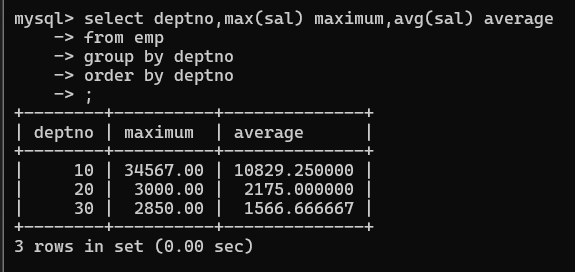
13. find how many employess have not received commission



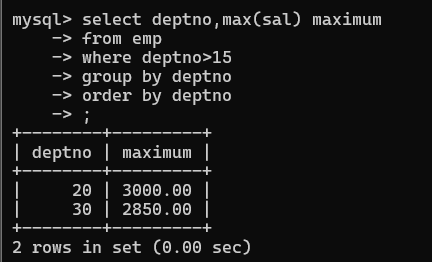
14. find sum of sal of all employees working in dept no 10



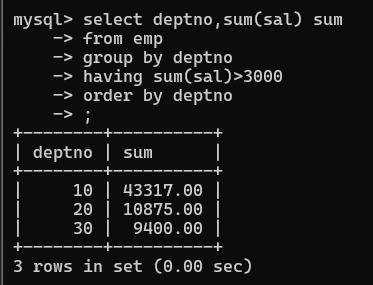
15. find maximum salary,average sal for each job in every department



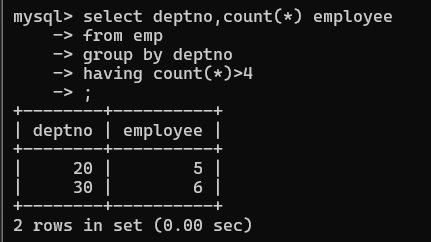
16. find max salary for every department if deptno is > 15 and arrange data in deptno order.



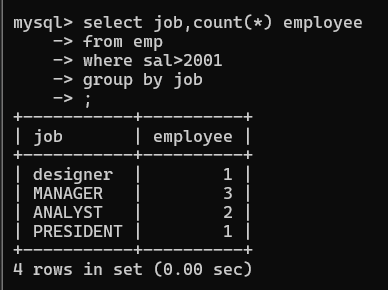
17. find sum salary for every department if sum is > 3000



18. list all department which has minimum 5 employees



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



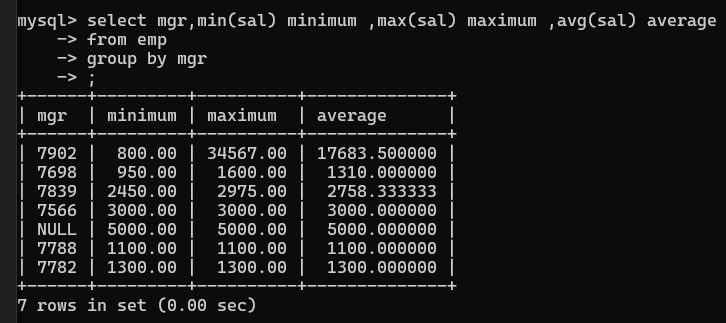
20. list all enames and jobs in small case letter



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

to left

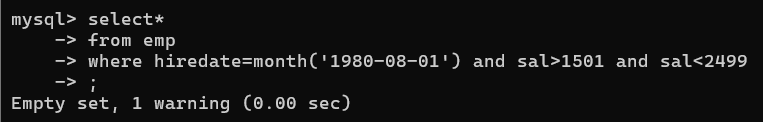
22. display min sal,max sal, average sal for all employees working under same manager



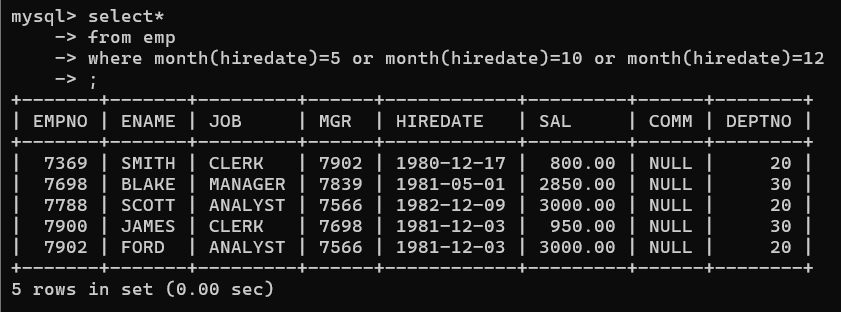
23. find sum of total earnings(sal+comm), average of sal+comm,

for all employees who earn sal > 2000 and work in either dept no 10 or 20

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

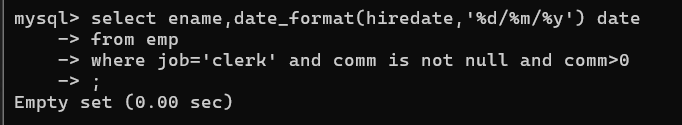


25. list all employees joined in either aug or may or dec



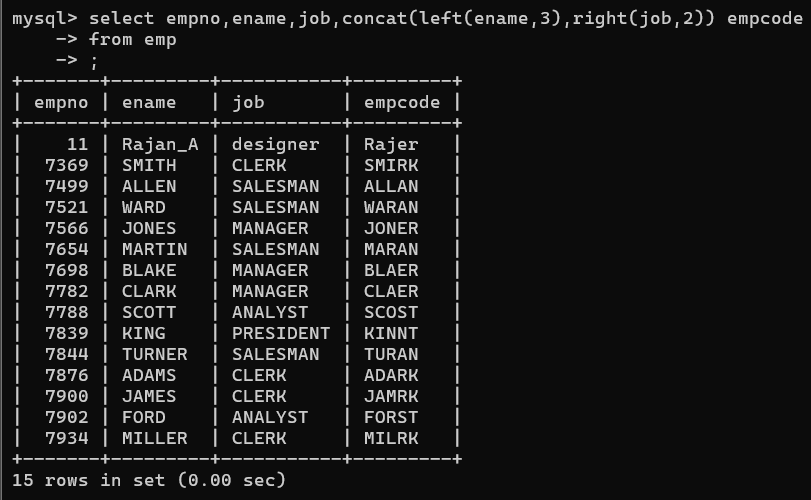
26. display name and hiredate in dd/mm/yy format for all employees whose job is clerk and they

earn some commission



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)



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

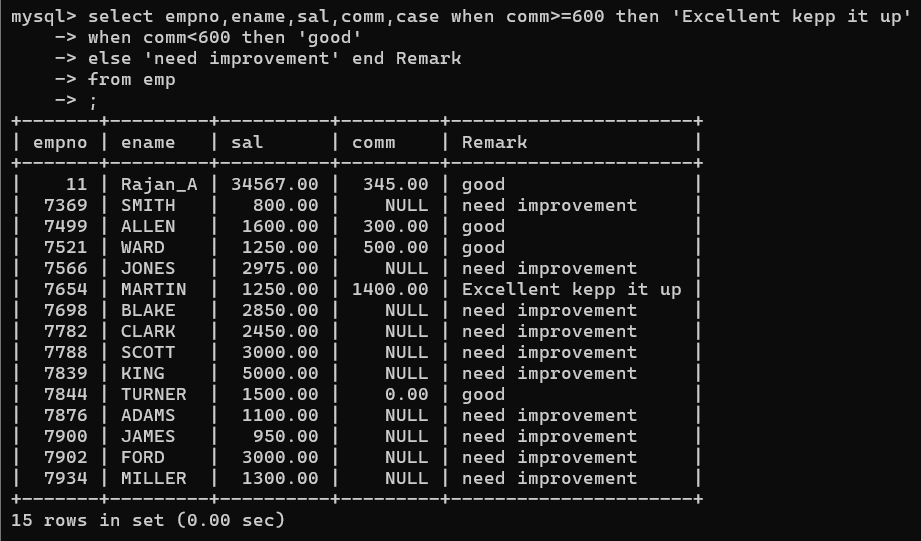
employees whose name starts with A and ends with N

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

comm >= 600 "excellent Keep it up"

if it < 600 or not null "good"

otherwise "Need improvement"



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



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

31. Practice creating following tables

MySQL syntax:

create table mydept\_DBDA

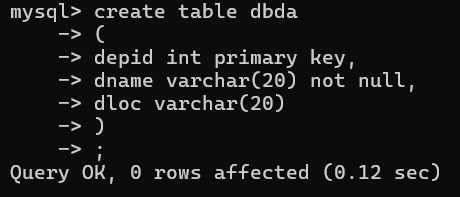
(

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),

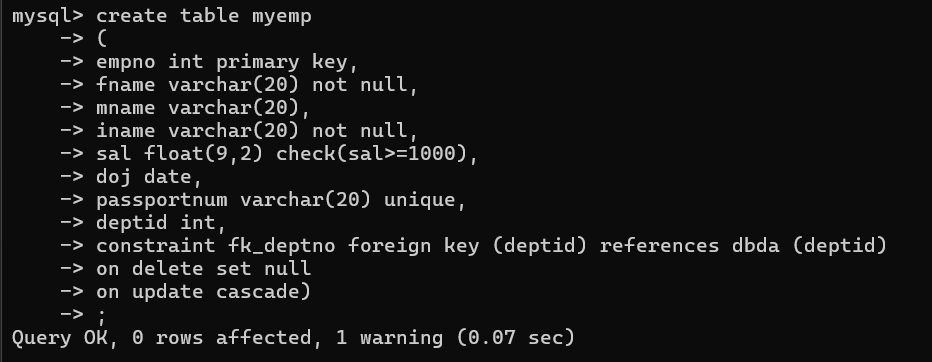
doj date default sysdate,

passportnum varchar2(15) unique,

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

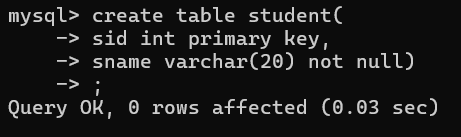
cascade

)

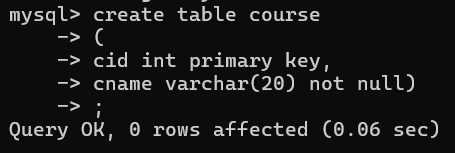


32. Create following tables Student, Course

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,

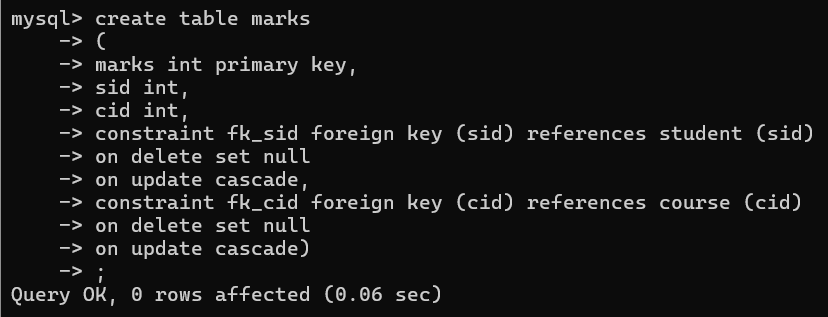
marks number,

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

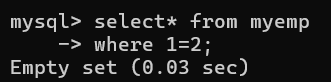
(

select \*

from emp

where 1=2;

)



34. Solve following using alter table

add primary key constraint on emp,dept,salgrade

emp ----→ empno

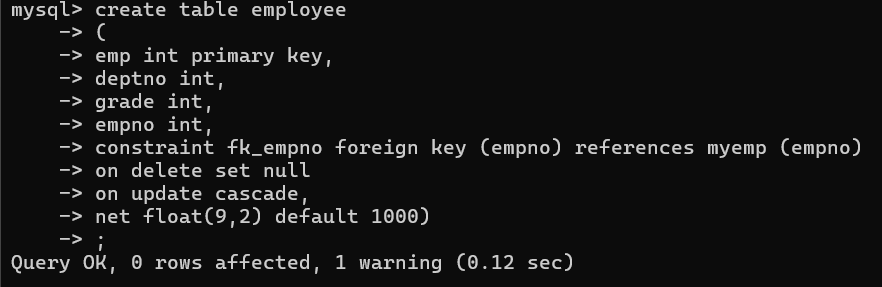
dept---→ deptno

salgrade---→ grade

add foreign key constarint in emp

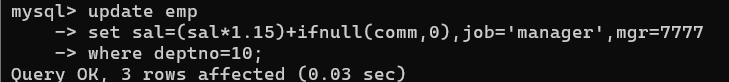
deptno --->> dept(deptno)

add new column in emp table netsal with constraint default 1000

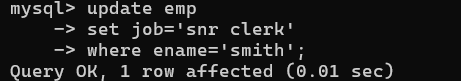


35. Update employee sal ---- increase sal of each employee by 15 % sal +comm, change the job to

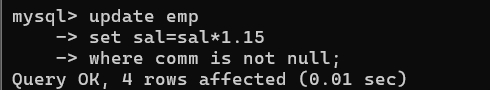
manager and mgr to 7777 for all employees in deptno 10.



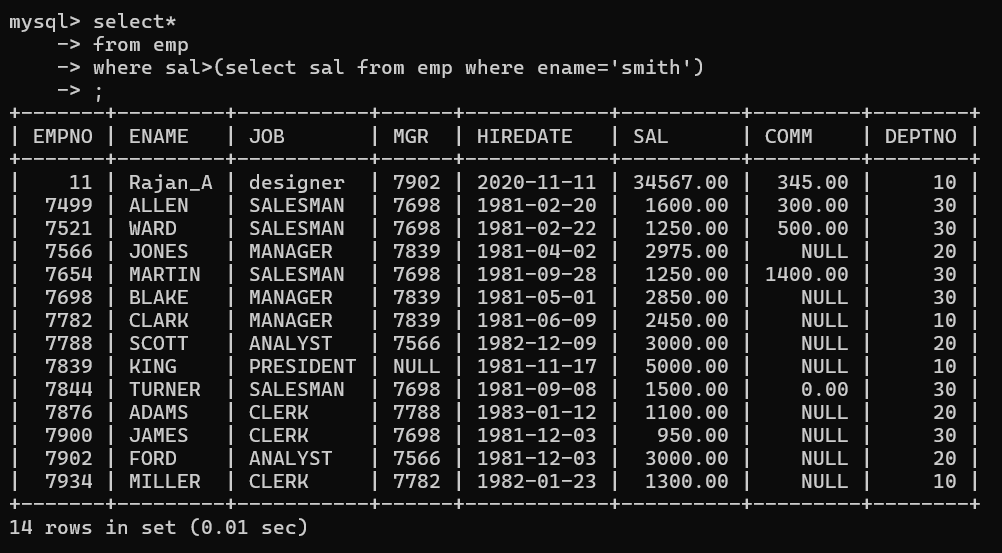
36. change job of smith to senior clerk



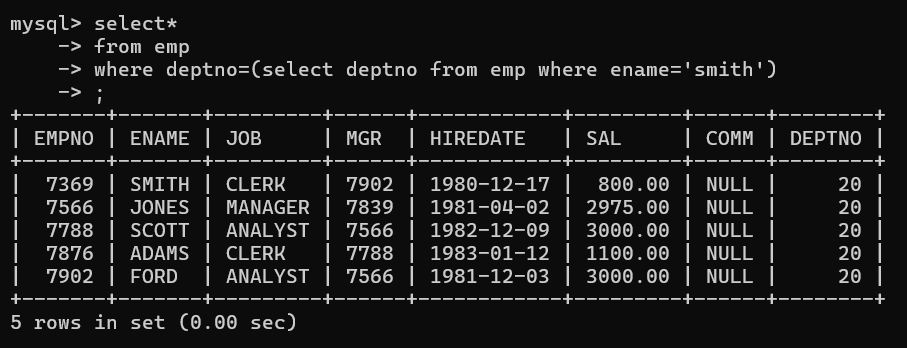
37. increase salary of all employees by 15% if they are earning some commission



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



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



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

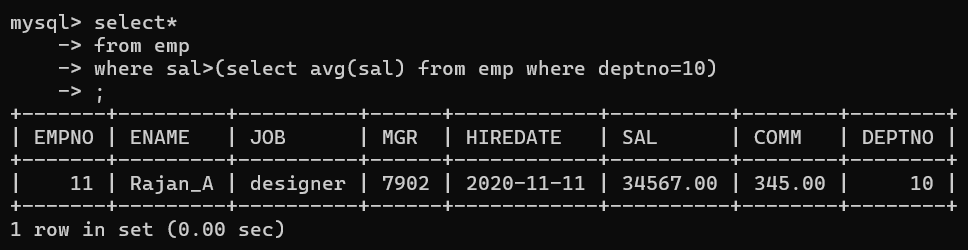
41. delete all employees working in alan's department

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

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

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

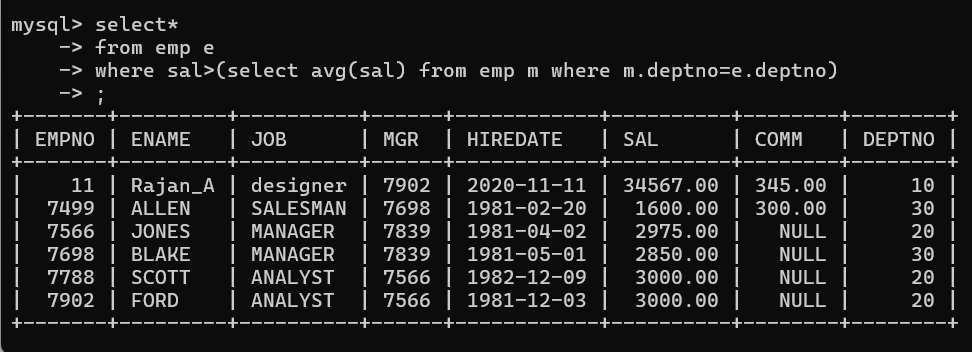
45. list all employees who earn more than average sal of dept 10



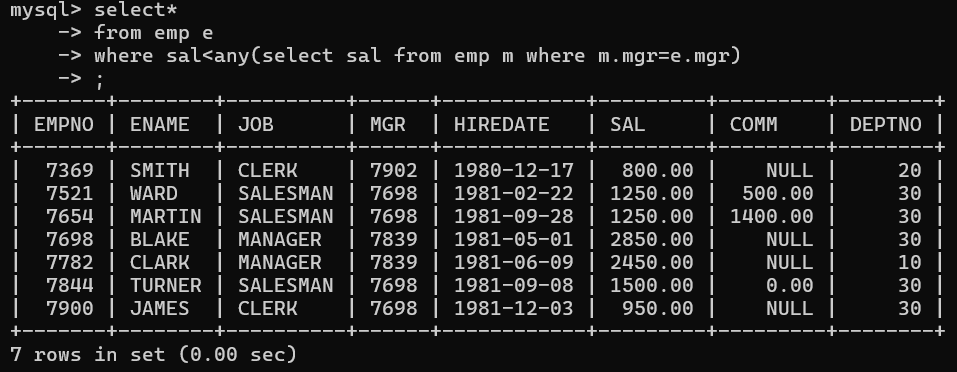
46. list all employees who earn more than average sal of Alan's department

47. list all employees who are working in purchase department

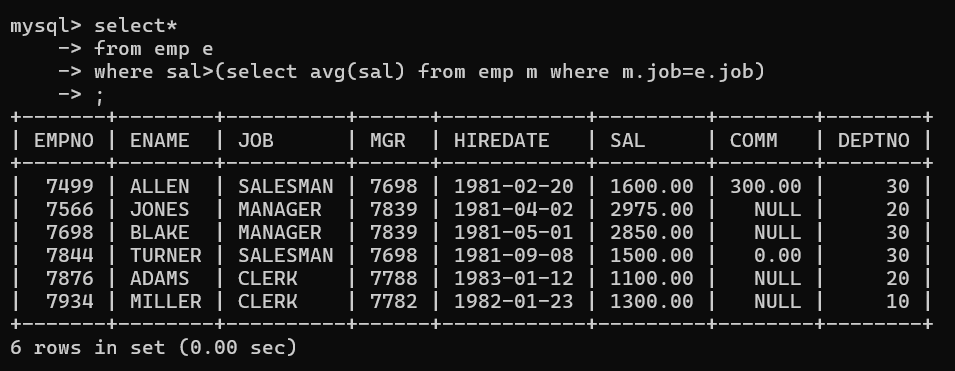
48. list all employees who earn more than average salary of their own department



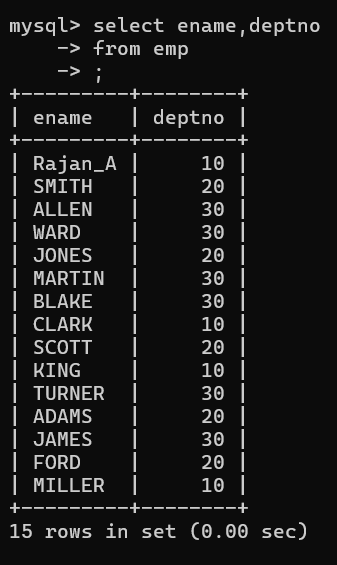
49. list all employees who earn sal < than their managers salary



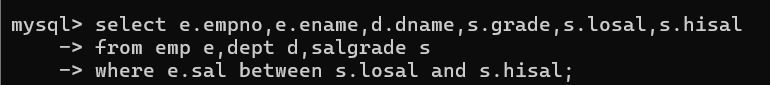
50. list all employees who are earning more than average salary of their job



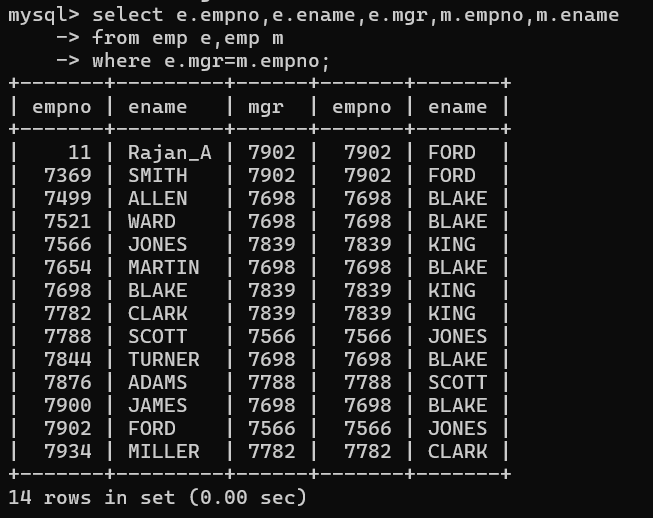
51. display employee name and department



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



sel53. list all employees number,name, mgrno and manager name



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)

create table product(

pid int primary key,

pname varchar(20) not null,

price float(9,2),

qty int,

ccid int,

ssid int,

constraint fk\_ccid foreign key (ccid) references category (cid)

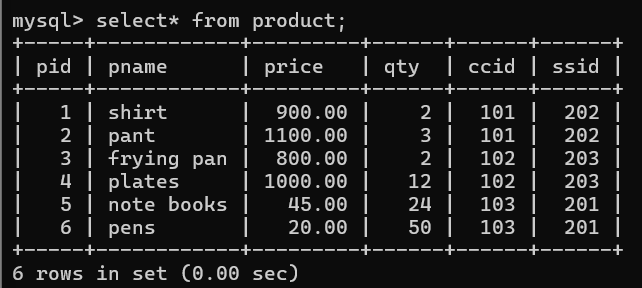
on delete set null

on update cascade,

constraint fk\_ssid foreign key (ssid) references salesman (sid)

on delete set null

on update cascade);



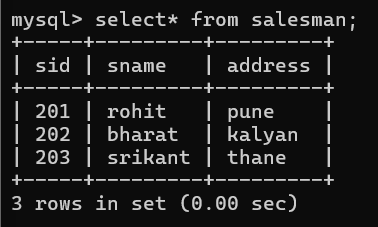
salesman (sid,sname,address)

create table category(

cid int primary key,

cname varchar(20),

description varchar(20));



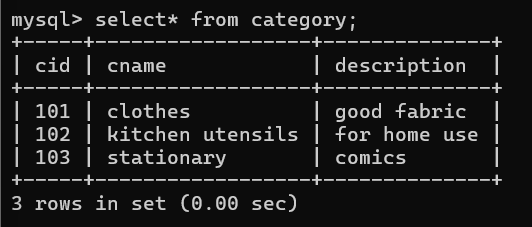
category(cid,cnam,descritpion)

create table salesman(

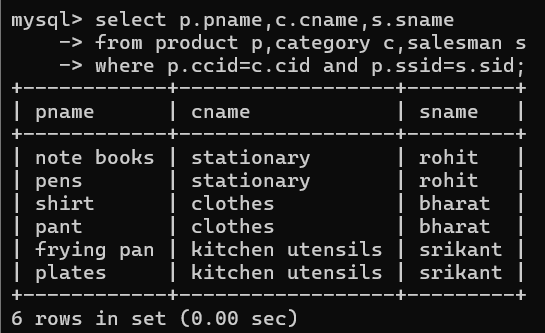
sid int primary key,

sname varchar(20) not null,

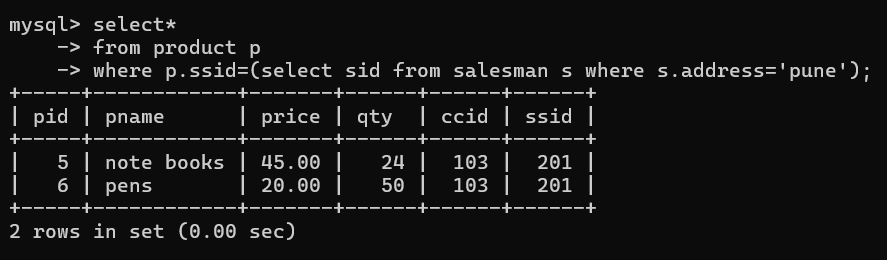
address varchar(50) not null);



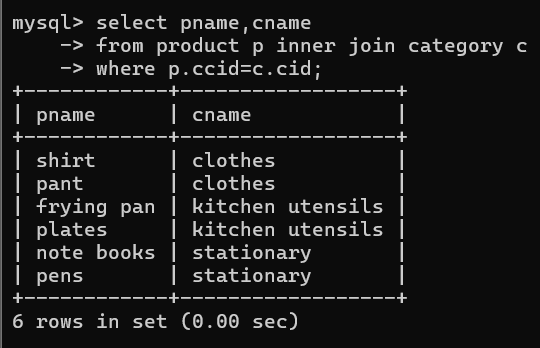
1. list all product name,their category name and name of a person, who sold that product



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



3. list all product name and category name



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

foreign keys are marked in green

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

courses(cid,cname,rid,fid)

room(roomid,rname,rloc)

faculty

fid fname spskill1 spskill2

10 kjzhcjhz a b

11 sdd x z

12 lksjk a x

13 ksdjlkj a b

courses

cid cname rid fid

121 DBDA 100 10

131 DAC 101

141 DTISS

151 DIOT 105 12

Room

roomid rname rloc

100 jasmin 1st floor

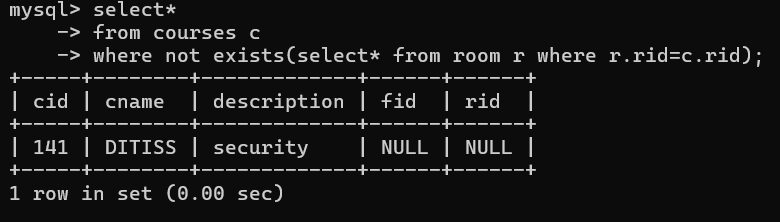
101 Rose 2nd floor

105 Lotus 1st floor

103 Mogra 1st floor

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

Available



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

allocated to any course

select null,null,f.fid,f.fname,null,c.cname,c.fid

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

-> where c.fid is null

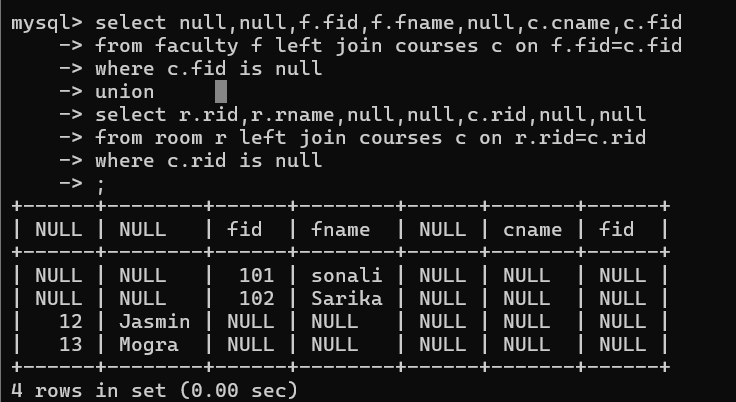
-> union

-> select r.rid,r.rname,null,null,c.rid,null,null

-> from room r left join courses c on r.rid=c.rid

-> where c.rid is null

-> ;



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

select r.rname,c.cname

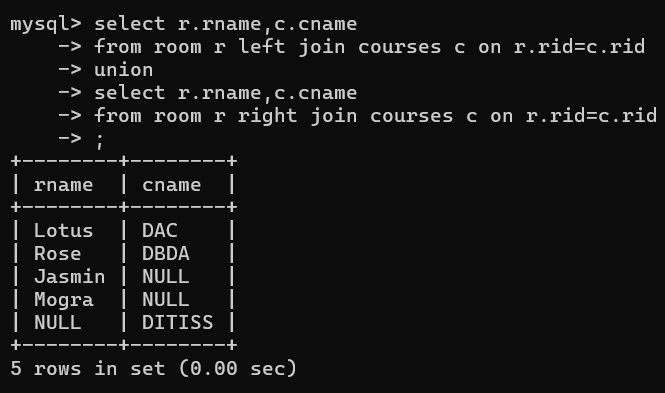
-> from room r left join courses c on r.rid=c.rid

-> union

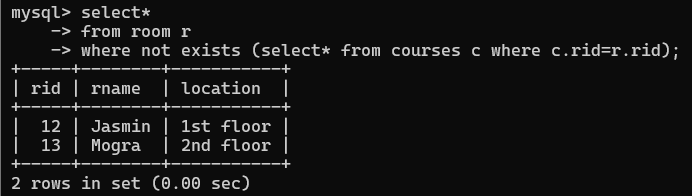
-> select r.rname,c.cname

-> from room r right join courses c on r.rid=c.rid

-> ;

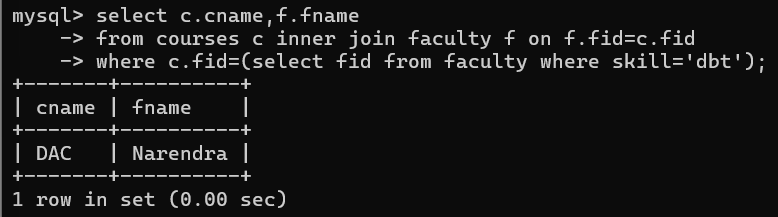


4. list all rooms which are not allocated to any courses



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

create table product(

-> pid int primary key,

-> pname varchar(20),

-> qty int,

-> price float(9,2),

-> cid 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 saleman (sid)

-> on delete set null

-> on update cascade);

insert into product values(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),(125,'pringles',40,60.00,1,14);

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

create table saleman(

-> sid int primary key,

-> sname varchar(20),

-> city varchar(20));

insert into saleman values(11,'Rahul','Pune'),

-> (12,'Kirti','Mumbai'),(13,'Prasad','Nashik'),

-> (14,'Arnav','Amravati');

11 Rahul Pune

12 Kirti Mumbai

13 Prasad Nashik

14 Arnav Amaravati

category ---- cname unique and not null

cid cname description

create table category(

-> cid int primary key,

-> cname varchar(20) not null,

-> description varchar(50));

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

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

-> (3,'snacks','yummy'),(4,'cold drink','thanda thanda cool cool');

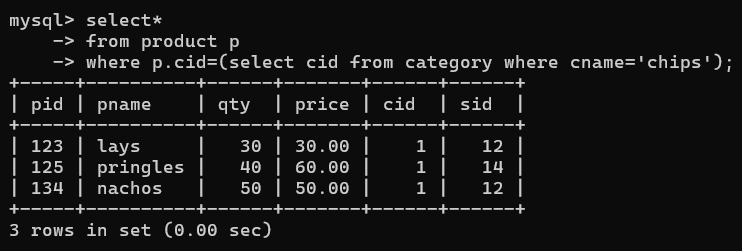
1 chips very crunchy

2 chocolate very chocolaty

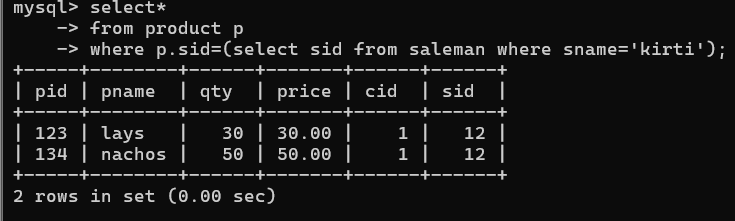
3 snacks yummy

4 cold drinks thanda thanda cool cool

1. List all products with category chips



2. display all products sold by 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