1. **Insurance DB:-**

create database insurancedb;

use insurancedb;

CREATE TABLE PERSON(

driver\_id varchar(10),

name varchar(20),

address varchar(15),

primary key(driver\_id));

CREATE TABLE CAR(

regno varchar(10),

model varchar(20),

Year int,

primary key(regno));

CREATE TABLE ACCIDENT(

report\_no int,

adate date,

location varchar(15),

primary key(report\_no));

CREATE TABLE OWNS(

driver\_id varchar(10),

regno varchar(10),

primary key(driver\_id,regno),

foreign key(driver\_id) references PERSON(driver\_id) on delete cascade,

foreign key(regno) references CAR(regno) on delete cascade);

CREATE TABLE PARTICIPATED(

driver\_id varchar(10),

regno varchar(10),

report\_no int,

damage\_amt float,

foreign key(driver\_id,regno) references OWNS(driver\_id,regno) on delete cascade,

foreign key(report\_no) references ACCIDENT(report\_no)on delete cascade);

show tables;

insert into PERSON values("1111","Ramu","K,S Layout");

insert into PERSON values("2222","John","Indiranagar");

insert into PERSON values("3333","Priya","Jayanagar");

insert into PERSON values("4444","Gopal","Whilefield");

insert into PERSON values("5555","Latha","Vijaynagar");

insert into CAR values("KA04Q2301","Maruthi-dx",2000);

insert into CAR values("KA05P1000","Fordicon",2000);

insert into CAR values("KA03L1234","Zen-VXI",1999);

insert into CAR values("KA03L9999","Maruthi-DX",2002);

insert into CAR values("KA01P4020","Indica-VX",2002);

insert into ACCIDENT values(12,"2002-05-01","M G Road");

insert into ACCIDENT values(200,"2002-12-10","Double Road");

insert into ACCIDENT values(300,"1999-07-23","M G Road");

insert into ACCIDENT values(25000,"2000-06-11","Residency Road");

insert into ACCIDENT values(26500,"2001-10-16","Richmond Road");

insert into OWNS values("1111","KA04Q2301");

insert into OWNS values("1111","KA05P1000");

insert into OWNS values("2222","KA03L1234");

insert into OWNS values("3333","KA03L9999");

insert into OWNS values("4444","KA01P4020");

insert into PARTICIPATED values("1111","KA04Q2301",12,20000);

insert into PARTICIPATED values("2222","KA03L1234",200,500);

insert into PARTICIPATED values("3333","KA03L9999",300,10000);

insert into PARTICIPATED values("4444","KA01P4020",25000,2375);

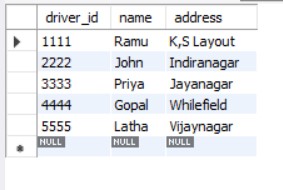
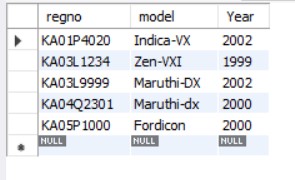
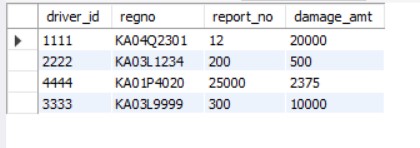
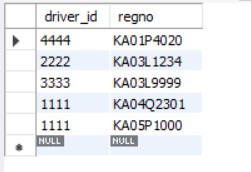
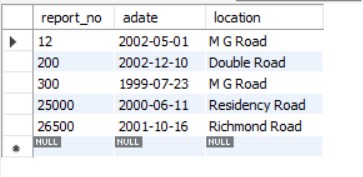
insert into PARTICIPATED values("2222","KA03L9999",12,10000);

update PARTICIPATED set damage\_amt=25000 where report\_no=12 and regno="KA03Q2301";

select count(\*) from ACCIDENT where adate like "2002-\_\_-\_\_";

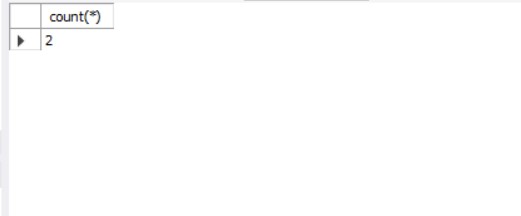
select count(a.report\_no) from ACCIDENT A, PARTICIPATED P, CAR C

where A.report\_no=P.report\_no and P.regno=c.regno and C.model="Maruthi-DX";

**Tables : **

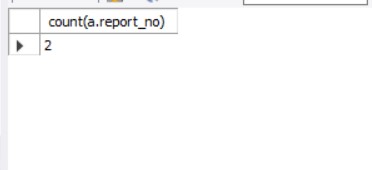
**Output:**

select count(\*) from ACCIDENT where adate like "2002-\_\_-\_\_";



select count(a.report\_no) from ACCIDENT A, PARTICIPATED P, CAR C

where A.report\_no=P.report\_no and P.regno=c.regno and C.model="Maruthi-DX";



1. **Book Dealer DB:-**

create database bookdealerdb;

use bookdealerdb;

CREATE TABLE author(

author\_id int,

name varchar(20),

city varchar(20),

country varchar(20),

primary key(author\_id)

);

CREATE TABLE publisher(

publisher\_id int,

name varchar(20),

city varchar(20),

country varchar(20),

primary key(publisher\_id)

);

CREATE TABLE catalog(

book\_id int,

title varchar(20),

author\_id int,

publisher\_id int,

category\_id int,

year int,

price int,

primary key(book\_id),

foreign key(author\_id) references author(author\_id),

foreign key(publisher\_id) references publisher(publisher\_id),

foreign key(category\_id) references category(category\_id)

);

CREATE TABLE category(

category\_id int,

description varchar(30),

primary key(category\_id)

);

CREATE TABLE order\_details(

order\_no int,

book\_id int,

quantity int,

primary key(order\_no),

foreign key(book\_id) references catalog(book\_id) on delete cascade

);

insert into author values(1001,"Teras Chan","CA","USA");

insert into author values(1002,"Stevens","Zombi","Uganda");

insert into author values(1003,"M Mano","Cair","Canada");

insert into author values(1004,"Karthik B P","New York","USA");

insert into author values(1005,"Willian Stalling","Las Vegas","USA");

insert into publisher values(1,"Pearson","New York","USA");

insert into publisher values(2,"EEE","New South Vales","USA");

insert into publisher values(3,"PHI","Delhi","India");

insert into publisher values(4,"Willey","Berlin","Germany");

insert into publisher values(5,"MGH","New York","USA");

insert into category values(1001,"Computer Science");

insert into category values(1002,"Algorithm Design");

insert into category values(1003,"Electronics");

insert into category values(1004,"Programming");

insert into category values(1005,"Operating Systems");

insert into catalog values(11,"Unix System",1001,1,1001,2000,251);

insert into catalog values(12,"Digital Signals",1002,2,1003,2001,425);

insert into catalog values(13,"Login Design",1003,3,1002,1999,225);

insert into catalog values(14,"Server Prg",1004,4,1004,2001,333);

insert into catalog values(15,"Linux OS",1005,5,1005,2003,326);

insert into catalog values(16,"C++ Bible",1005,5,1001,2000,526);

insert into catalog values(17,"Cobol Handbook",1005,4,1001,2000,658);

insert into order\_details values(1,11,5);

insert into order\_details values(2,12,8);

insert into order\_details values(3,13,15);

insert into order\_details values(4,14,22);

insert into order\_details values(5,15,3);

insert into order\_details values(6,17,10);

select \* from author;

select \* from publisher;

select \* from category;

select \* from catalog;

select \* from order\_details;

use bookdealerdb;

/\*--details of the authors having aleast 2 books along with year of publish is greater than 2000\*/

select \* from author where author\_id in(select author\_id from catalog where year>=2000 group by author\_id having count(author\_id)>=2);

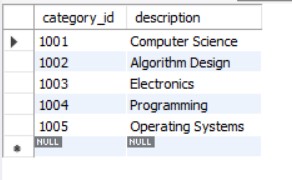
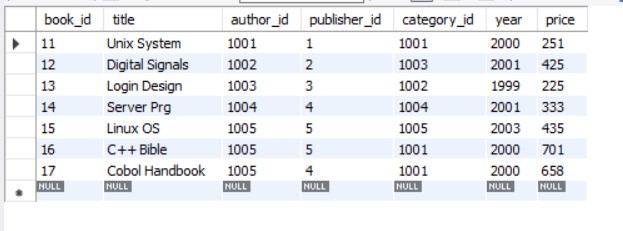
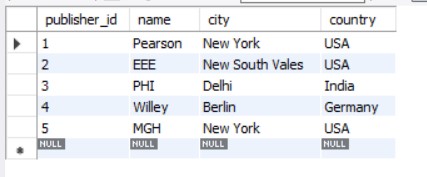
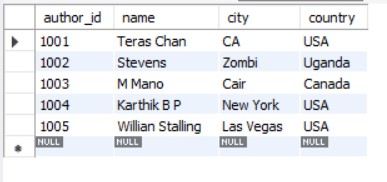
/\*--author names with max sales of books\*/

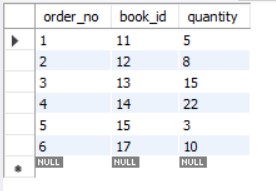
select a.name from author a, catalog c, order\_details o where a.author\_id=c.author\_id and c.book\_id=o.book\_id and o.quantity=(select max(quantity) from order\_details);

/\*--increase the price of book by 10% of a specified publisher(eg:- publisher\_id=5)\*/

update catalog set price=(price+price\*0.1) where publisher\_id=5;

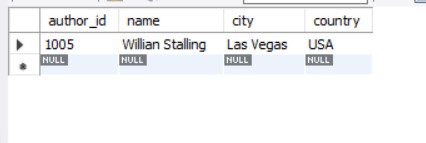
**Tables:**



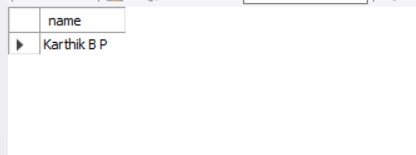


**Output:**

select \* from author where author\_id in(select author\_id from catalog where year>=2000 group by author\_id having count(author\_id)>=2);



select a.name from author a, catalog c, order\_details o where a.author\_id=c.author\_id and c.book\_id=o.book\_id and o.quantity=(select max(quantity) from order\_details);



1. **Order DB:**

create database orderdb;

use orderdb;

create table CUSTOMER(

cust\_no int,

cname varchar(20),

city varchar(20),

primary key(cust\_no)

);

create table ordertb(

order\_no int,

odate date,

cust\_no int,

order\_amount int,

primary key(order\_no),

foreign key(cust\_no) references CUSTOMER(cust\_no)

);

create table item(

item\_no int,

unit\_price int,

primary key(item\_no)

);

create table ORDER\_ITEM(

order\_no int,

item\_no int,

qty int,

foreign key(order\_no) references ordertb(order\_no),

foreign key(item\_no) references item(item\_no),

primary key(order\_no,item\_no)

);

create table warehouse(

warehouse\_no int,

city varchar(20),

primary key(warehouse\_no)

);

create table shipment(

order\_no int,

warehouse\_no int,

odate date,

foreign key(order\_no) references ordertb(order\_no),

foreign key(warehouse\_no) references warehouse(warehouse\_no),

primary key(order\_no,warehouse\_no)

);

insert into customer values(771,"Pushpa K","Bangalore");

insert into customer values(772,"Suman","Mumbai");

insert into customer values(773,"Sourav","Calicut");

insert into customer values(774,"Laila","Hyderabad");

insert into customer values(775,"Faizal","Bangalore");

insert into ordertb values(111,"2002-01-22",771,18000);

update ordertb set order\_no=111 where order\_no="111";

insert into ordertb values(112,"2002-07-30",774,6000);

insert into ordertb values(113,"2003-04-03",775,9000);

insert into ordertb values(114,"2003-11-03",775,29000);

insert into ordertb values(115,"2002-12-10",773,29000);

insert into ordertb values(116,"2004-08-19",772,56000);

insert into ordertb values(117,"2004-09-10",771,20000);

insert into ordertb values(118,"2004-11-20",775,29000);

insert into ordertb values(119,"2005-02-13",774,29000);

insert into ordertb values(120,"2005-10-13",775,29000);

insert into item values(5001,503);

insert into item values(5002,750);

insert into item values(5003,150);

insert into item values(5004,600);

insert into item values(5005,890);

insert into order\_item values(111,5001,50);

insert into order\_item values(112,5003,20);

insert into order\_item values(113,5002,50);

insert into order\_item values(114,5005,60);

insert into order\_item values(115,5004,90);

insert into order\_item values(116,5001,10);

insert into order\_item values(117,5003,80);

insert into order\_item values(118,5005,50);

insert into order\_item values(119,5002,10);

insert into order\_item values(120,5004,45);

insert into warehouse values(1,"Delhi");

insert into warehouse values(2,"Mumbai");

insert into warehouse values(3,"Chennai");

insert into warehouse values(4,"Bangalore");

insert into warehouse values(5,"Bangalore");

insert into warehouse values(6,"Delhi");

insert into warehouse values(7,"Mumbai");

insert into warehouse values(8,"Chennai");

insert into warehouse values(9,"Delhi");

insert into warehouse values(10,"Bangalore");

insert into shipment values(111,1,"2002-02-10");

insert into shipment values(112,5,"2002-09-10");

insert into shipment values(113,8,"2003-02-10");

insert into shipment values(114,3,"2003-12-10");

insert into shipment values(115,9,"2004-01-19");

insert into shipment values(116,1,"2004-09-20");

insert into shipment values(117,5,"2004-09-10");

insert into shipment values(118,7,"2004-11-30");

insert into shipment values(119,7,"2005-04-30");

insert into shipment values(120,6,"2005-12-21");

use orderdb;

select \* from customer;

select \* from ordertb;

select \* from item;

select \* from order\_item;

select \* from warehouse;

select \* from shipment;

use orderdb;

/\*Question 1\*/

select c.cname,count(o.order\_no),avg(o.order\_amount) from customer c

inner join ordertb o

on c.cust\_no=o.cust\_no

group by c.cust\_no;

/\*Question 2\*/

select o.order\_no,w.city from ordertb o

inner join shipment s

on o.order\_no=s.order\_no

inner join warehouse w

on s.warehouse\_no=w.warehouse\_no

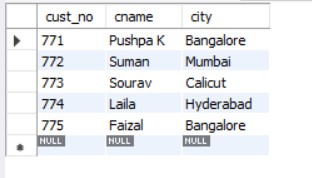
where w.city="Delhi" /\*This can be added to specify the city to be viewed\*/

order by w.city;

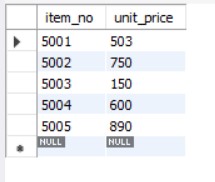
/\*Question 3\*/

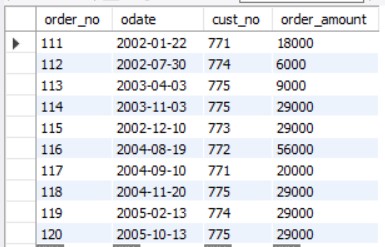
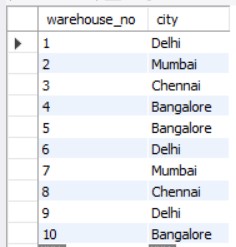
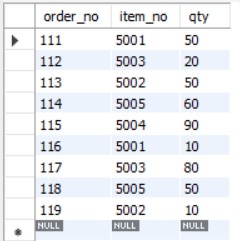
delete from order\_item where order\_no=120;

**Tables:**









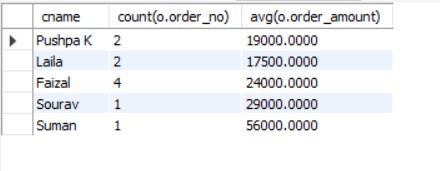
**Outputs:**

select c.cname,count(o.order\_no),avg(o.order\_amount) from customer c

inner join ordertb o

on c.cust\_no=o.cust\_no

group by c.cust\_no;



select o.order\_no,w.city from ordertb o

inner join shipment s

on o.order\_no=s.order\_no

inner join warehouse w

on s.warehouse\_no=w.warehouse\_no

where w.city="Delhi" /\*This can be added to specify the city to be viewed\*/

order by w.city;



1. **Banking DB:**

create database bankingdb;

use bankingdb;

CREATE TABLE branch

( branch\_name VARCHAR(15),

branch\_city VARCHAR(15),

assets float,

PRIMARY KEY(branch\_name)

);

CREATE TABLE account

( accno INT,

branch\_name VARCHAR(15),

balance float,

PRIMARY KEY(accno),

FOREIGN KEY(branch\_name) REFERENCES branch(branch\_name)ON DELETE CASCADE

);

CREATE TABLE customer

( customer\_name VARCHAR(15),

customer\_street VARCHAR(15),

customer\_city VARCHAR(15),

PRIMARY KEY(customer\_name)

);

CREATE TABLE loan

( loan\_number INT,

branch\_name VARCHAR(15),

amount float,

PRIMARY KEY(loan\_number),

FOREIGN KEY(branch\_name) REFERENCES branch(branch\_name)

);

CREATE TABLE depositor

( customer\_name VARCHAR(15),

accno INTEGER,

PRIMARY KEY(customer\_name, accno),

FOREIGN KEY(customer\_name) REFERENCES customer(customer\_name),

FOREIGN KEY(accno) REFERENCES account(accno)

);

CREATE TABLE borrower

( customer\_name VARCHAR(15),

loan\_number INT,

PRIMARY KEY(customer\_name, loan\_number),

FOREIGN KEY(customer\_name) REFERENCES customer(customer\_name),

FOREIGN KEY(loan\_number) REFERENCES loan(loan\_number)

);

INSERT INTO BRANCH VALUES

("SBI PD NAGAR", "BANGALORE", 200000),

("SBI RS NAGAR", "BANGALORE" ,500000),

("SBI JAYANAGAR" ,"CHENNAI",60000),

("SBI VIJAYNAGAR" ,"DELHI", 870000),

("SBI GBROAD" ,"DELHI", 550000);

INSERT INTO ACCOUNT VALUES

(1000,"SBI PD NAGAR",5000),

(1001,"SBI RS NAGAR",5000),

(1002,"SBI JAYANAGAR",5000),

(1003,"SBI VIJAYNAGAR",40000),

(1004,"SBI GBROAD" ,4000);

INSERT INTO ACCOUNT VALUES

(1005,"SBI GBROAD",5500),

(1006,"SBI VIJAYNAGAR",2500);

INSERT INTO CUSTOMER VALUES

("RAM","BSTREET 45","BANGALORE"),

("SHYAM","BSTREET 46","BANGALORE"),

("RAJAT","CSTREET 55","CHENNAI"),

("AAKASH","DSTREET 65","DELHI"),

("SURESH","DSTREET 66","DELHI");

INSERT INTO DEPOSITOR VALUES

("RAM",1000),

("SHYAM",1001),

("RAJAT",1002),

("AAKASH",1003),

("SURESH",1004);

INSERT INTO DEPOSITOR VALUES

("SURESH",1005),

("AAKASH",1006);

INSERT INTO LOAN VALUES

(10,"SBI PD NAGAR",10000),

(11,"SBI RS NAGAR",20000),

(12,"SBI JAYANAGAR",30000),

(13,"SBI VIJAYNAGAR",40000),

(14,"SBI GBROAD" ,50000);

INSERT INTO BORROWER VALUES

("RAM",10),

("SHYAM",11),

("RAJAT",12),

("AAKASH",13),

("SURESH",14);

/\*QUES 1\*/

SELECT customer\_name

FROM depositor d,account a

WHERE d.accno=a.accno

/\*AND a.branch\_name='SBI VIJAYNAGAR'\*//\*THIS CAN BE ADDED TO SPECIFY BRANCH NAME\*/

GROUP BY d.customer\_name

HAVING COUNT(d.customer\_name)>=2;

/\*QUES 2\*/

SELECT d.customer\_name FROM account a,branch b,depositor d

WHERE b.branch\_name=a.branch\_name AND

a.accno=d.accno AND

b.branch\_city='CHENNAI'

GROUP BY d.customer\_name

HAVING COUNT(distinct b.branch\_name)=(

SELECT COUNT(branch\_name)

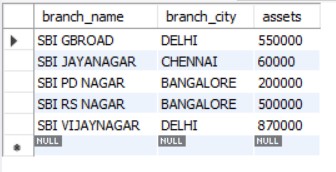
FROM branch

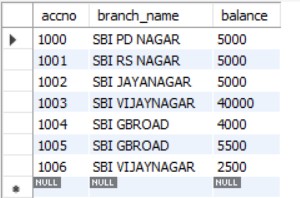
WHERE branch\_city='CHENNAI');

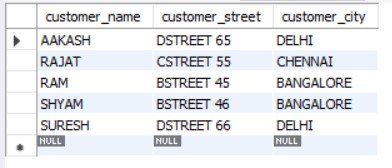
/\*QUES 3\*/

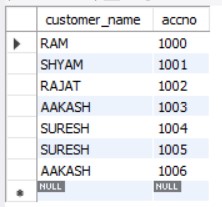
DELETE FROM account WHERE branch\_name IN(SELECT branch\_name FROM branch WHERE branch\_city='CHENNAI');

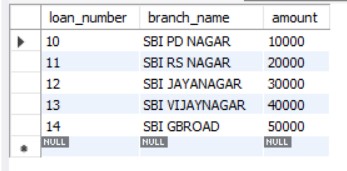
**Tables:**

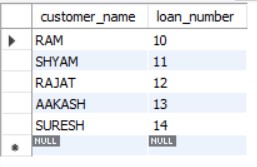












**Outputs:**

SELECT customer\_name

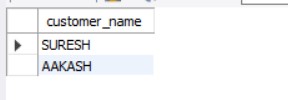
FROM depositor d,account a

WHERE d.accno=a.accno

/\*AND a.branch\_name='SBI VIJAYNAGAR'\*/ /\*THIS CAN BE ADDED TO SPECIFY BRANCH NAME\*/

GROUP BY d.customer\_name

HAVING COUNT(d.customer\_name)>=2;



SELECT d.customer\_name FROM account a,branch b,depositor d

WHERE b.branch\_name=a.branch\_name AND

a.accno=d.accno AND

b.branch\_city='CHENNAI'

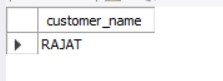
GROUP BY d.customer\_name

HAVING COUNT(distinct b.branch\_name)=(

SELECT COUNT(branch\_name)

FROM branch

WHERE branch\_city='CHENNAI');



1. **Student Enrollment DB:-**

CREATE DATABASE STUDENT\_ENROLLMENT;

USE STUDENT\_ENROLLMENT;

CREATE TABLE STUDENT(

REGNO VARCHAR(10),

NAME VARCHAR(20),

MAJOR VARCHAR(5),

BDATE DATE,

PRIMARY KEY(REGNO)

);

CREATE TABLE COURSE(

COURSE\_NO INT,

CNAME VARCHAR(5),

DEPT VARCHAR(5),

PRIMARY KEY(COURSE\_NO)

);

CREATE TABLE ENROLL(

REGNO VARCHAR(10),

COURSE\_NO INT,

MARKS INT,

SEM INT,

PRIMARY KEY(REGNO, COURSE\_NO),

FOREIGN KEY(REGNO) REFERENCES STUDENT(REGNO),

FOREIGN KEY(COURSE\_NO) REFERENCES COURSE(COURSE\_NO)

);

CREATE TABLE BOOK\_ADOPTION(

COURSE\_NO INT,

SEM INT,

BOOK\_ISBN INT,

FOREIGN KEY(COURSE\_NO) REFERENCES COURSE(COURSE\_NO),

FOREIGN KEY(BOOK\_ISBN) REFERENCES TEXTBOOK(BOOK\_ISBN)

);

CREATE TABLE TEXTBOOK(

BOOK\_ISBN INT,

BOOKTITLE VARCHAR(50),

PUBLISHER VARCHAR(20),

AUTHOR VARCHAR(20),

PRIMARY KEY(BOOK\_ISBN)

);

INSERT INTO STUDENT VALUES

("CS01","RAM","DS","1986-03-12"),

("IS02","SMITH","USP","1987-12-21"),

("EC03","AHMED","SNS","1985-04-17"),

("CS03","SNEHA","DBMS","1987-01-01"),

("TC05","AKHILA","EC","1986-10-06");

INSERT INTO COURSE VALUES

(11,"DS","CS"),

(22,"USP","IS"),

(33,"SNS","EC"),

(44,"DBMS","CS"),

(55,"EC","TC");

INSERT INTO ENROLL VALUES

("CS01",11,4,85),

("IS02",22,6,80),

("EC03",33,2,80),

("CS03",44,6,75),

("TC05",55,2,8);

INSERT INTO TEXTBOOK VALUES

(1,"DS AND C","PRINCETON","PADMA REDDY"),

(2,"FUNDAMENTALS OF C","PRINCETON","GODSE"),

(3,"FUNDAMENTALS OF DBMS","PRINCETON","NAVATHE"),

(4,"SQL","PRINCETON","FOLEY"),

(5,"ELECTRONIC CIRCUITS","TMH","ELMASRI"),

(6,"ADV UNIX PROG","TMH","STEVENS");

INSERT INTO BOOK\_ADOPTION VALUES

(11,4,1),

(11,4,2),

(44,6,3),

(44,6,4),

(55,2,5),

(22,6,6);

DELETE FROM BOOK\_ADOPTION

WHERE BOOK\_ISBN=7;

DELETE FROM TEXTBOOK

WHERE BOOK\_ISBN=7;

/\*QUES 1\*/

INSERT INTO TEXTBOOK VALUES(7,"JAVA THE COMPLETE REFERENCE","ORACLE PRESS","HERBERT SCHILDT");

INSERT INTO BOOK\_ADOPTION VALUES(55,2,7);

/\*QUES 2\*/

SELECT C.COURSE\_NO,BA.BOOK\_ISBN, TB.BOOKTITLE FROM COURSE C, BOOK\_ADOPTION BA, TEXTBOOK TB

WHERE C.COURSE\_NO=BA.COURSE\_NO AND BA.BOOK\_ISBN=TB.BOOK\_ISBN AND C.DEPT="CS"

AND 2<=(SELECT COUNT(BOOK\_ISBN) FROM BOOK\_ADOPTION B

WHERE C.COURSE\_NO=B.COURSE\_NO)

ORDER BY TB.BOOKTITLE;

/\*QUES 3\*/

SELECT DISTINCT C.DEPT FROM COURSE C

INNER JOIN BOOK\_ADOPTION BA

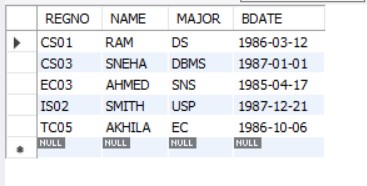
ON BA.COURSE\_NO=C.COURSE\_NO

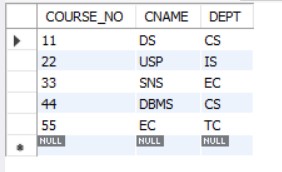
INNER JOIN TEXTBOOK TB

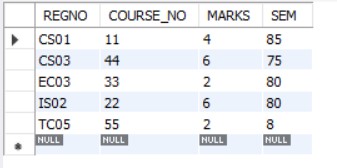
ON TB.BOOK\_ISBN=BA.BOOK\_ISBN

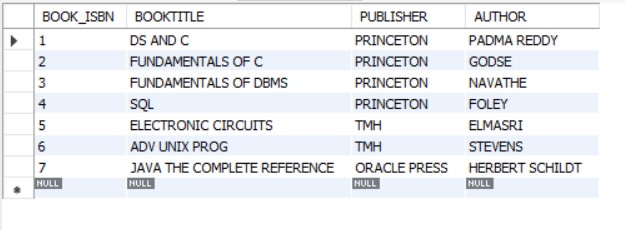
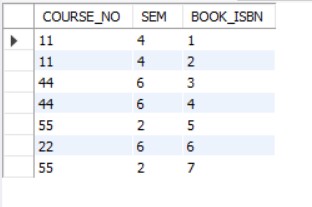
WHERE TB.PUBLISHER="PRINCETON";

**Tables:**









**Outputs:**

SELECT DISTINCT C.DEPT FROM COURSE C

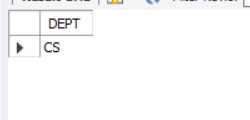
INNER JOIN BOOK\_ADOPTION BA

ON BA.COURSE\_NO=C.COURSE\_NO

INNER JOIN TEXTBOOK TB

ON TB.BOOK\_ISBN=BA.BOOK\_ISBN

WHERE TB.PUBLISHER="PRINCETON";



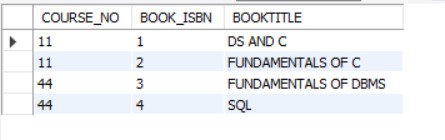
SELECT C.COURSE\_NO,BA.BOOK\_ISBN, TB.BOOKTITLE FROM COURSE C, BOOK\_ADOPTION BA, TEXTBOOK TB

WHERE C.COURSE\_NO=BA.COURSE\_NO AND BA.BOOK\_ISBN=TB.BOOK\_ISBN AND C.DEPT="CS"

AND 2<=(SELECT COUNT(BOOK\_ISBN) FROM BOOK\_ADOPTION B

WHERE C.COURSE\_NO=B.COURSE\_NO)

ORDER BY TB.BOOKTITLE;



**6)Movies DB:-**

create database moviedb;

use moviedb;

create table actor(

act\_id int,

act\_name varchar(20),

act\_gender varchar(3),

primary key(act\_id)

);

create table director(

dir\_id int,

dir\_name varchar(20),

dir\_phone int,

primary key(dir\_id)

);

create table movies(

mov\_id int,

mov\_title varchar(30),

mov\_year int,

mov\_lang varchar(20),

dir\_id int,

primary key(mov\_id),

foreign key(dir\_id) references director(dir\_id) on delete cascade on update cascade

);

create table movie\_cast(

act\_id int,

mov\_id int,

role varchar(20),

primary key(act\_id,mov\_id),

foreign key(act\_id) references actor(act\_id) on delete cascade on update cascade,

foreign key(mov\_id) references movies(mov\_id) on delete cascade on update cascade

);

create table rating (

mov\_id int,

rev\_stars real,

foreign key(mov\_id) references movies(mov\_id) on delete cascade on update cascade

);

insert into actor values

(100,"Steve Carrel","M"),

(101,"John Krasinski","M"),

(102,"Jenna Fischer","F"),

(103,"Ed Helms","M"),

(104,"Mindy Kaling","F");

insert into actor values

(105,"Jamie Bell","M"),

(106,"Tom Hanks","M"),

(107,"Christian Bale","M"),

(108,"Laura Dern","F");

insert into director values

(1,"Greg Daniels",459876),

(2,"Steven Spielberg",465462),

(3,"Tom Shadyac",465423),

(4,"Christopher Nolan",898434),

(5,"David Lynch",134657);

insert into movies values

(1000,"The Office",2005,"English",1),

(1001,"TinTin",2011,"English",2),

(1002,"The Terminal",2004,"English",2),

(1003,"The Dark Night",2008,"English",4),

(1004,"Evan Almighty",2007,"English",3),

(1005,"Blue Velvet",1986,"English",5);

insert into movie\_cast values

(100,1000,"Michael Scott"),

(105,1001,"TinTin"),

(106,1002,"Victor"),

(107,1003,"Bruce Wayne"),

(100,1004,"Evan Baxter"),

(108,1005,"Sandy Williams");

insert into rating values

(1000,8.8),

(1001,7.5),

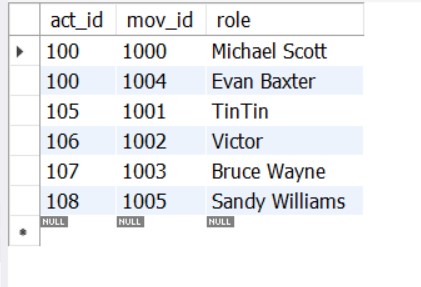
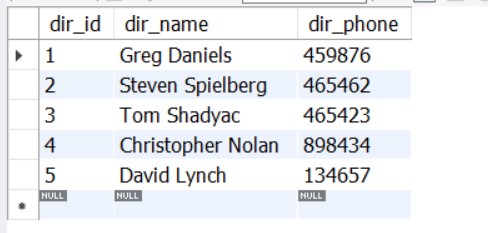
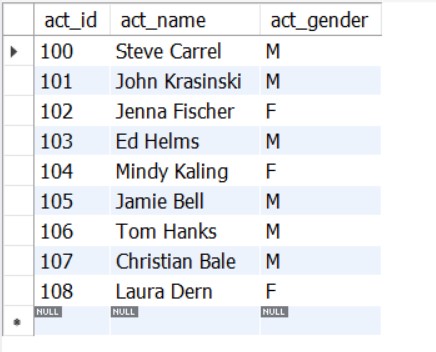
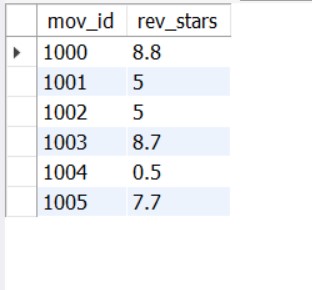
(1002,8.9),

(1003,8.7),

(1004,0.5),

(1005,7.7);

**Tables:-**



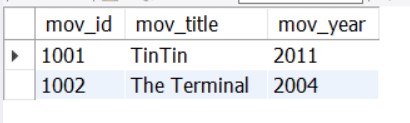
**Outputs:-**

/\* Ques 1 \*/

select m.mov\_id,m.mov\_title,m.mov\_year from movies m

inner join director d

on d.dir\_id=m.dir\_id

where dir\_name="Steven Spielberg";

/\* Ques 2 \*/

select m.mov\_id,m.mov\_title,m.mov\_year from movie\_cast mc

inner join movies m

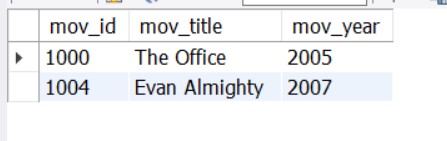
on m.mov\_id=mc.mov\_id

where mc.act\_id=(

select mci.act\_id from movie\_cast mci

group by mci.act\_id

having count(mci.act\_id)>=2

);

/\* Ques 3 \*/

select a.act\_name,m.mov\_title from movie\_cast mc

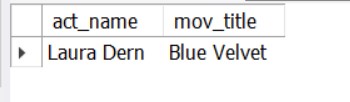
inner join actor a

on a.act\_id=mc.act\_id

inner join movies m

on mc.mov\_id=m.mov\_id

where m.mov\_year>=2015 or m.mov\_year<=2000;



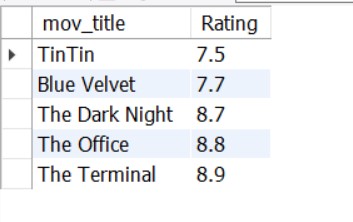
/\* Ques 4 \*/

select m.mov\_title,r.rev\_stars as "Rating" from movies m

inner join rating r

on r.mov\_id=m.mov\_id

where r.rev\_stars>=1

order by r.rev\_stars;

/\* Ques 5 \*/

update rating r

set r.rev\_stars= 5

where mov\_id=any(

select distinct m.mov\_id from movies m

inner join director d

on d.dir\_id=m.dir\_id

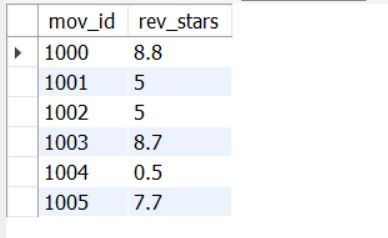
where d.dir\_name="Steven Spielberg"

);

update rating r

set r.rev\_stars=8.9

where r.mov\_id=1002;



**7) AirlinesDB**

create database airlinedb;

use airlinedb;

create table flights(

flno int,

from\_loc varchar(20),

to\_loc varchar(20),

distance int,

departs time,

arrives time,

price int,

primary key(flno)

);

create table aircraft(

aid int,

aname varchar(20),

cruisingrange int,

primary key(aid)

);

create table certified(

eid int,

aid int,

foreign key(aid) references aircraft(aid),

foreign key(eid) references employee(eid),

primary key(aid,eid)

);

create table employee(

eid int,

ename varchar(20),

salary int,

primary key(eid)

);

insert into flights values

(1000,"Chicago","Tulsa",268,"22:30:00","01:15:00",500),

(1001,"Madison","Albuquerque",500,"12:30:00","14:00:00",100),

(1002,"Albuquerque","New York",450,"14:30:00","16:00:00",300),

(1003,"Madison","Philadelphia",400,"12:30:00","15:00:00",450),

(1004,"Philadelphia","New York",120,"16:10:00","17:30:00",200),

(1005,"New York","Nashville",670,"19:30:00","01:00:00",100),

(1006,"El Paso","Santa Fe",300,"05:20:00","09:15:00",350);

insert into flights values

(1007,"Bangalore","Frankfurt",10000,"10:30:00","22:00:00",2000),

(1008,"Bangalore","Frankfurt",10000,"01:20:00","10:45:00",1500),

(1009,"Bangalore","Frankfurt",10000,"06:55:00","23:35:00",2000);

insert into flights values

(1010,"Bangalore","Chennai",236,"06:55:00","23:35:00",200),

(1011,"Chennai","New Delhi",2387,"06:55:00","23:35:00",400),

(1012,"Bangalore","Mumbai",800,"06:55:00","23:35:00",100),

(1013,"Mumbai","New Delhi",780,"06:55:00","23:35:00",150);

insert into flights values

(1014,"Bangalore","New Delhi",1800,"06:55:00","23:35:00",120);

update flights

set flights.distance=1800

where flights.flno=1014;

update flights

set flights.to\_loc="New Delhi"

where flights.flno=1014;

insert into aircraft values

(100,"Boeing 747-400",2000),

(101,"Boeing 777-200",3000),

(102,"Boeing 777-300",1500),

(103,"AirBus A380-800",4000),

(104,"AirBus A380-plus",5000),

(105,"Airbus A350-500",3700),

(106,"Fly High-500",1000),

(107,"Cruise Perfect-200",1700),

(108,"Cruise Perfect-500",4000),

(109,"Fly High-400",7800);

insert into employee values

(10,"Alex",10000),

(11,"Sam",14000),

(12,"Jonas",17500),

(13,"John",10600),

(14,"Michael",9000),

(15,"Stanley",2000),

(16,"Chris",4000),

(17,"Verstappen",16000),

(18,"Perez",13000),

(19,"Hamilton",1300);

insert into employee values

(20,"Norris",12300),

(21,"Riccardo",13000),

(22,"Lincoln",18000),

(23,"Cole",14000);

insert into employee values

(24,"Ryan",600),

(25,"Jan",800),

(26,"Gareth",1100);

insert into certified values

(10,100),

(11,101),

(12,104),

(13,103),

(14,104),

(20,105),

(20,106),

(20,107),

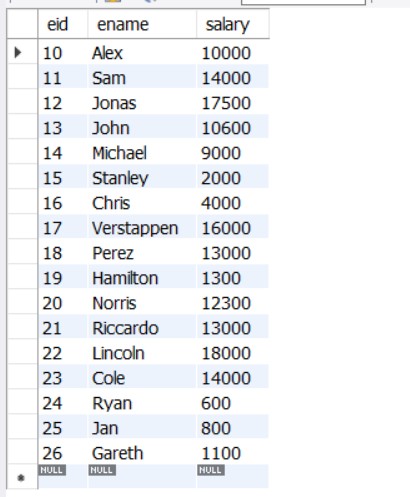
(21,108),

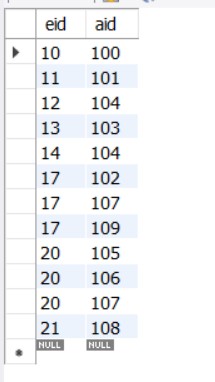
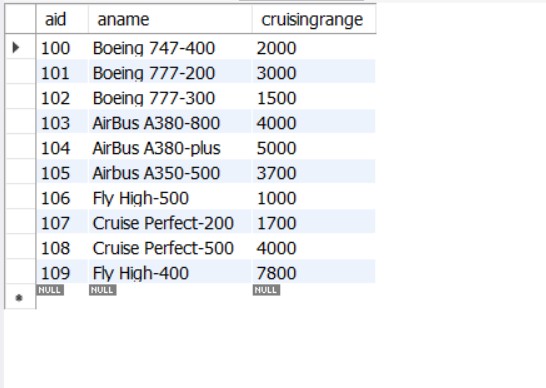
(17,102),

(17,109),

(17,107);

**Tables:-**





**Outputs:-**

/\* QUES 1 (changed the salary from Rs.80,000 to $14,000)\*/

select a.aname from certified c

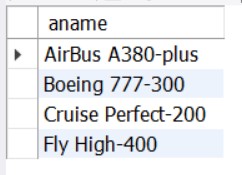
inner join aircraft a

on a.aid=c.aid

inner join employee e

on e.eid=c.eid

where e.salary>14000;



/\* QUES 2 \*/

select e.eid,max(a.cruisingrange) from certified c

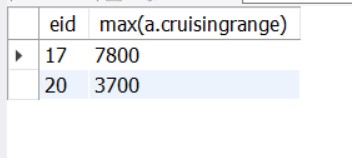
inner join employee e

on e.eid=c.eid

inner join aircraft a

on a.aid=c.aid

group by c.eid having count(c.eid)>=3;



/\* QUES 3 \*/

select e.ename from employee e

where e.salary<(select min(f.price) from flights f

where from\_loc="Bangalore"and to\_loc="Frankfurt");



/\* Ques 4 (Changed the cruising range from 1000 to 3000 for conveniece)\*/

select a.aname, avg(e.salary) from certified c

inner join aircraft a

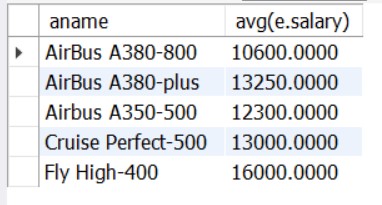
on a.aid=c.aid

inner join employee e

on e.eid=c.eid

where a.cruisingrange>3000

group by a.aid;



/\* QUES 5 \*/

select e.ename, a.aname from certified c

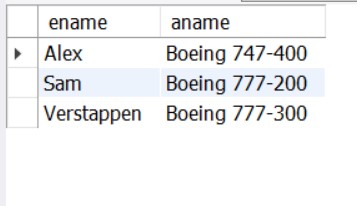
inner join aircraft a

on a.aid=c.aid

inner join employee e

on e.eid=c.eid

where a.aname like "Boeing%";

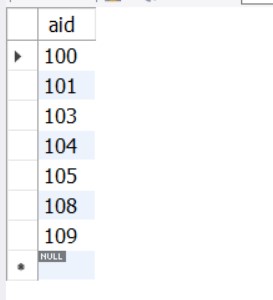


/\*QUES 6\*/

select a.aid from aircraft a

where a.cruisingrange>=(select f.distance from flights f

where from\_loc="Bangalore" and to\_loc="New Delhi");



**8)StudentDB**

|  |
| --- |
| create database student\_faculty; |
|  |

|  |
| --- |
| use student\_faculty; |
|  |

|  |
| --- |
| create table student ( |
|  |

|  |
| --- |
| usn varchar (10) primary key, |
|  |

|  |
| --- |
| sname varchar (25), |
|  |

|  |
| --- |
| address varchar (25), |
|  |

|  |
| --- |
| phone int (10), |
|  |

|  |
| --- |
| gender char (1)); |
|  |

|  |
| --- |
| create table semsec ( |
|  |

|  |
| --- |
| ssid varchar (5) primary key, |
|  |

|  |
| --- |
| sem int (2), |
|  |

|  |
| --- |
| sec char (1)); |
|  |

|  |
| --- |
| create table class ( |
|  |

|  |
| --- |
| usn varchar (10), |
|  |

|  |
| --- |
| ssid varchar (5), primary |
|  |

|  |
| --- |
| key (usn, ssid), |
|  |

|  |
| --- |
| foreign key (usn) references student (usn), |
|  |

|  |
| --- |
| foreign key (ssid) references semsec (ssid)); |
|  |

|  |
| --- |
| create table subject ( |
|  |

|  |
| --- |
| subcode varchar (8), |
|  |

|  |
| --- |
| title varchar (20), |
|  |

|  |
| --- |
| sem int (2), |
|  |

|  |
| --- |
| credits int (2), |
|  |

|  |
| --- |
| primary key (subcode)); |
|  |

|  |
| --- |
| create table iamarks ( |
|  |

|  |
| --- |
| usn varchar (10), |
|  |

|  |
| --- |
| subcode varchar (8), |
|  |

|  |
| --- |
| ssid varchar(5), |
|  |

|  |
| --- |
| test1 int(2), |
|  |

|  |
| --- |
| test2 int(2), |
|  |

|  |
| --- |
| test3 int(2), |
|  |

|  |
| --- |
| finalia int (2), |
|  |

|  |
| --- |
| primary key (usn, subcode, ssid), |
|  |

|  |
| --- |
| foreign key (usn) references student (usn), |
|  |

|  |
| --- |
| foreign key (subcode) references subject (subcode), |
|  |

|  |
| --- |
| foreign key (ssid) references semsec (ssid)); |
|  |

|  |
| --- |
| insert into student values('1rn13cs020','akshay','belagavi', |
|  |

|  |
| --- |
| 88778811,'m'); |
|  |

|  |
| --- |
| insert into student values('1rn13cs062','sandhya','bengaluru', |
|  |

|  |
| --- |
| 77228299,'f'); |
|  |

|  |
| --- |
| insert into student values('1rn13cs091','teesha','bengaluru', |
|  |

|  |
| --- |
| 77123123,'f'); |
|  |

|  |
| --- |
| insert into student values('1rn13cs066','supriya','mangaluru', |
|  |

|  |
| --- |
| 88778811,'f'); |
|  |

|  |
| --- |
| insert into student values('1rn14cs010','abhay','bengaluru', |
|  |

|  |
| --- |
| 99002112,'m'); |
|  |

|  |
| --- |
| insert into student values('1rn14cs032','bhaskar','bengaluru', |
|  |

|  |
| --- |
| 99232110,'m'); |
|  |

|  |
| --- |
| insert into student values ('1rn14cs025','asmi','bengaluru', 78947373,'f'); |
|  |

|  |
| --- |
| insert into student values ('1rn15cs011','ajay','tumkur', 98450913,'m'); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| insert into student values ('1rn15cs029','chitra','davangere', |
|  |

|  |
| --- |
| 76967721,'f'); |
|  |

|  |
| --- |
| insert into student values ('1rn15cs045','jeeva','bellary', 99448501,'m'); |
|  |

|  |
| --- |
| insert into student values ('1rn15cs091','santosh','mangaluru', |
|  |

|  |
| --- |
| 8812332,'m'); |
|  |

|  |
| --- |
| insert into student values('1rn16cs045','ismail','kalburgi', |
|  |

|  |
| --- |
| 99002322,'m'); |
|  |

|  |
| --- |
| insert into student values ('1rn16cs088','sameera','shimoga', |
|  |

|  |
| --- |
| 99055422,'f'); |
|  |

|  |
| --- |
| insert into student values ('1rn16cs122','vinayaka','chikamagalur', |
|  |

|  |
| --- |
| 88008800,'m'); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| insert into semsec values ('cse8a', 8,'a'); |
|  |

|  |
| --- |
| insert into semsec values ('cse8b', 8,'b'); |
|  |

|  |
| --- |
| insert into semsec values ('cse8c',8,'c'); |
|  |

|  |
| --- |
| insert into semsec values ('cse7a',7,'a'); |
|  |

|  |
| --- |
| insert into semsec values ('cse7b',7,'b'); |
|  |

|  |
| --- |
| insert into semsec values ('cse7c',7,'c'); |
|  |

|  |
| --- |
| insert into semsec values ('cse6a',6,'a'); |
|  |

|  |
| --- |
| insert |
|  |

|  |
| --- |
| into semsec values ('cse6b', 6,'b'); |
|  |

|  |
| --- |
| insert into semsec values ('cse6c', 6,'c'); |
|  |

|  |
| --- |
| insert into semsec values ('cse5a', 5,'a'); |
|  |

|  |
| --- |
| insert into semsec values ('cse5b', 5,'b'); |
|  |

|  |
| --- |
| insert into semsec values ('cse5c', 5,'c'); |
|  |

|  |
| --- |
| insert into semsec values ('cse4a',4,'a'); |
|  |

|  |
| --- |
| insert into semsec values ('cse4b', 4,'b'); |
|  |

|  |
| --- |
| insert into semsec values('cse4c',4,'c'); |
|  |

|  |
| --- |
| insert into semsec values ('cse3a', 3,'a'); |
|  |

|  |
| --- |
| insert into semsec values ('cse3b', 3,'b'); |
|  |

|  |
| --- |
| insert into semsec values('cse3c',3,'c'); |
|  |

|  |
| --- |
| insert into semsec values ('cse2a', 2,'c'); |
|  |

|  |
| --- |
| insert into semsec values ('cse2b', 2,'b'); |
|  |

|  |
| --- |
| insert into semsec values ('cse2c', 2,'c'); |
|  |

|  |
| --- |
| insert into semsec values ('cse1a', 1,'a'); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| insert into semsec values ('cse1b', 1,'b'); |
|  |

|  |
| --- |
| insert into semsec values ('cse1c', 1,'c'); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| insert into class values('1rn13cs020','cse8a'); |
|  |

|  |
| --- |
| insert into class values('1rn13cs062','cse8a'); |
|  |

|  |
| --- |
| insert into class values('1rn13cs066','cse8b'); |
|  |

|  |
| --- |
| insert into class values('1rn13cs091','cse8c'); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| insert into class values('1rn14cs010','cse7a'); |
|  |

|  |
| --- |
| insert into class values('1rn14cs025','cse7a'); |
|  |

|  |
| --- |
| insert into class values('1rn14cs032','cse7a'); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| insert into class values('1rn15cs011','cse4a'); |
|  |

|  |
| --- |
| insert into class values('1rn15cs029','cse4a'); |
|  |

|  |
| --- |
| insert into class values('1rn15cs045','cse4b'); |
|  |

|  |
| --- |
| insert into class values('1rn15cs091','cse4c'); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| insert into class values('1rn16cs045','cse3a'); |
|  |

|  |
| --- |
| insert into class values('1rn16cs088','cse3b'); |
|  |

|  |
| --- |
| insert into class values('1rn16cs122','cse3c'); |
|  |

|  |
| --- |
| insert into subject values ('10cs81','aca', 8, 4); |
|  |

|  |
| --- |
| insert into subject values ('10cs82','ssm', 8, 4); |
|  |

|  |
| --- |
| insert into subject values ('10cs83','nm', 8, 4); |
|  |

|  |
| --- |
| insert into subject values ('10cs84','cc', 8, 4); |
|  |

|  |
| --- |
| insert into subject values ('10cs85','pw', 8, 4); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| insert into subject values ('10cs71','ooad', 7, 4); |
|  |

|  |
| --- |
| insert into subject values ('10cs72','ecs', 7, 4); |
|  |

|  |
| --- |
| insert into subject values ('10cs73','ptw', 7, 4); |
|  |

|  |
| --- |
| insert into subject values ('10cs74','dwdm', 7, 4); |
|  |

|  |
| --- |
| insert into subject values ('10cs75','java', 7, 4); |
|  |

|  |
| --- |
| insert into subject values ('10cs76','san', 7, 4); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| insert into subject values ('15cs51', 'me', 5, 4); |
|  |

|  |
| --- |
| insert into subject values ('15cs52','cn', 5, 4); |
|  |

|  |
| --- |
| insert into subject values ('15cs53','dbms', 5, 4); |
|  |

|  |
| --- |
| insert into subject values ('15cs54','atc', 5, 4); |
|  |

|  |
| --- |
| insert into subject values ('15cs55','java', 5, 3); |
|  |

|  |
| --- |
| insert into subject values ('15cs56','ai', 5, 3); |
|  |

|  |
| --- |
| insert into subject values ('15cs41','m4', 4, 4); |
|  |

|  |
| --- |
| insert into subject values ('15cs42','se', 4, 4); |
|  |

|  |
| --- |
| insert into subject values ('15cs43','daa', 4, 4); |
|  |

|  |
| --- |
| insert into subject values ('15cs44','mpmc', 4, 4); |
|  |

|  |
| --- |
| insert into subject values ('15cs45','ooc', 4, 3); |
|  |

|  |
| --- |
| insert into subject values ('15cs46','dc', 4, 3); |
|  |

|  |
| --- |
| insert into subject values ('15cs31','m3', 3, 4); |
|  |

|  |
| --- |
| insert into subject values ('15cs32','ade', 3, 4); |
|  |

|  |
| --- |
| insert into subject values ('15cs33','dsa', 3, 4); |
|  |

|  |
| --- |
| insert into subject values ('15cs34','co', 3, 4); |
|  |

|  |
| --- |
| insert into subject values ('15cs35','usp', 3, 3); |
|  |

|  |
| --- |
| insert into subject values ('15cs36','dms', 3, 3); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| insert into iamarks (usn, subcode, ssid, test1, test2, test3)values |
|  |

|  |
| --- |
| ('1rn13cs091','10cs81','cse8c', 15, 16,18); |
|  |

|  |
| --- |
| insert into iamarks (usn, subcode, ssid, test1, test2, test3)values |
|  |

|  |
| --- |
| ('1rn13cs091','10cs82','cse8c', 12, 19,14); |
|  |

|  |
| --- |
| insert into iamarks (usn, subcode, ssid, test1, test2, test3)values |
|  |

|  |
| --- |
| ('1rn13cs091','10cs83','cse8c', 19, 15,20); |
|  |

|  |
| --- |
| insert into iamarks (usn, subcode, ssid, test1, test2, test3)values |
|  |

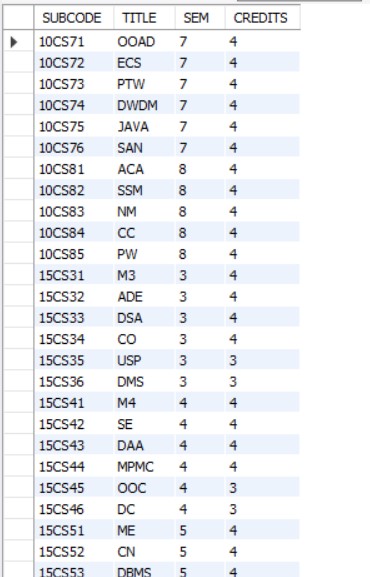
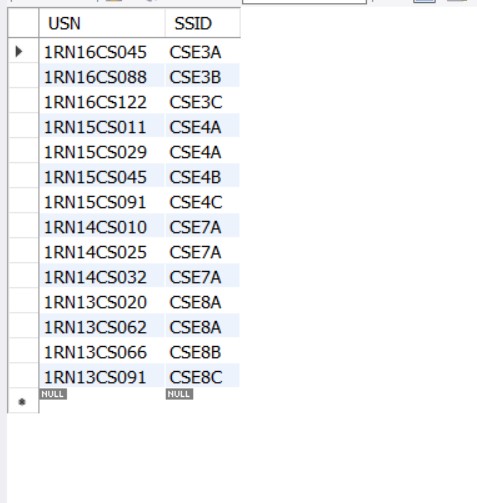
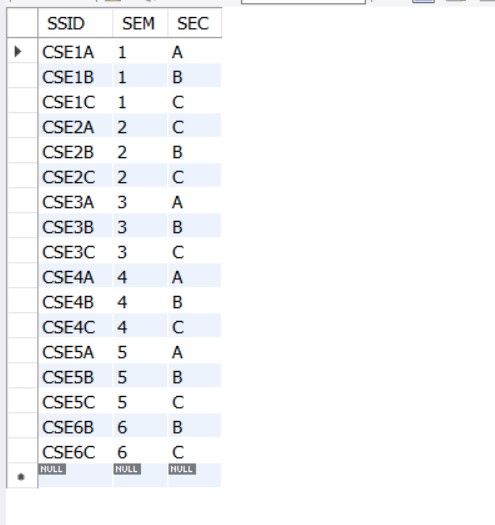
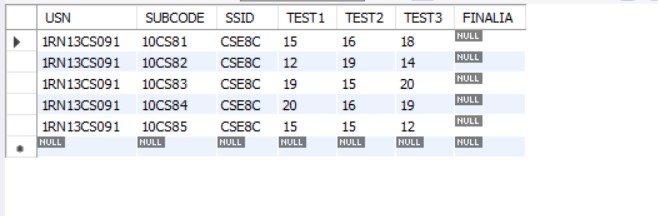
|  |
| --- |
| ('1rn13cs091','10cs84','cse8c', 20, 16,19); |
|  |

|  |
| --- |
| insert into iamarks (usn, subcode, ssid, test1, test2, test3)values |
|  |

|  |
| --- |
| ('1rn13cs091','10cs85','cse8c', 15, 15,12); |
|  |

|  |
| --- |
|  |
|  |

**Tables:**

****

**Outputs:**

|  |
| --- |
| select s.\*, ss.sem, ss.sec |
|  |

|  |
| --- |
| from student s, semsec ss, class c |
|  |

|  |
| --- |
| where s.usn = c.usn and |
|  |

|  |
| --- |
| ss.ssid = c.ssid and |
|  |

|  |
| --- |
| ss.sem = 4 and |
|  |

|  |
| --- |
| ss.sec='c'; |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| select ss.sem, ss.sec, s.gender, count(s.gender) as count |
|  |

|  |
| --- |
| from student s, semsec ss, class c |
|  |

|  |
| --- |
| where s.usn = c.usn and |
|  |

|  |
| --- |
| ss.ssid = c.ssid |
|  |

|  |
| --- |
| group by ss.sem, ss.sec, s.gender |
|  |

|  |
| --- |
| order by sem; |
|  |
|  |

|  |
| --- |
| create view stu\_test1\_marks\_view |
|  |

|  |
| --- |
| as |
|  |

|  |
| --- |
| select test1, subcode |
|  |

|  |
| --- |
| from iamarks |
|  |

|  |
| --- |
| where usn = ‘1rn13cs091’; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| select \* from stu\_test1\_marks\_view; |
|  |
|  |

|  |
| --- |
| -- query 4 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| delimiter // |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| create procedure avg\_marks() |
|  |

|  |
| --- |
| begin |
|  |

|  |
| --- |
| declare c\_a integer; |
|  |

|  |
| --- |
| declare c\_b integer; |
|  |

|  |
| --- |
| declare c\_c integer; |
|  |

|  |
| --- |
| declare c\_sum integer; |
|  |

|  |
| --- |
| declare c\_avg integer; |
|  |

|  |
| --- |
| declare c\_usn varchar(10); |
|  |

|  |
| --- |
| declare c\_subcode varchar(8); |
|  |

|  |
| --- |
| declare c\_ssid varchar(5); |
|  |

|  |
| --- |
| declare c\_iamarks cursor for |
|  |

|  |
| --- |
| select greatest(test1,test2) as a, greatest(test1,test3) as b, greatest(test3,test2) as c, usn, subcode, ssid |
|  |

|  |
| --- |
| from iamarks |
|  |

|  |
| --- |
| where finalia is null |
|  |

|  |
| --- |
| for update; |
|  |

|  |
| --- |
| open c\_iamarks; |
|  |

|  |
| --- |
| loop |
|  |

|  |
| --- |
| fetch c\_iamarks into c\_a, c\_b, c\_c, c\_usn, c\_subcode, c\_ssid; |
|  |

|  |
| --- |
| if (c\_a != c\_b) then |
|  |

|  |
| --- |
| set c\_sum=c\_a+c\_b; |
|  |

|  |
| --- |
| else |
|  |

|  |
| --- |
| set c\_sum=c\_a+c\_c; |
|  |

|  |
| --- |
| end if; |
|  |

|  |
| --- |
| set c\_avg=c\_sum/2; |
|  |

|  |
| --- |
| update iamarks set finalia = c\_avg |
|  |

|  |
| --- |
| where usn = c\_usn and subcode = c\_subcode and ssid = c\_ssid; |
|  |

|  |
| --- |
| end loop; |
|  |

|  |
| --- |
| close c\_iamarks; |
|  |

|  |
| --- |
| end; |
|  |

|  |
| --- |
| // |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| call avg\_marks(); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| select \* from iamarks; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| select \* from iamarks; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| -- query 5 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| select s.usn,s.sname,s.address,s.phone,s.gender, |
|  |

|  |
| --- |
| (case |
|  |

|  |
| --- |
| when ia.finalia between 17 and 20 then 'outstanding' |
|  |

|  |
| --- |
| when ia.finalia between 12 and 16 then 'average' |
|  |

|  |
| --- |
| else 'weak' |
|  |

|  |
| --- |
| end) as cat |
|  |

|  |
| --- |
| from student s, semsec ss, iamarks ia, subject sub |
|  |

|  |
| --- |
| where s.usn = ia.usn and |
|  |

|  |
| --- |
| ss.ssid = ia.ssid and |
|  |

|  |
| --- |
| sub.subcode = ia.subcode and |
|  |

sub.sem = 8;

**9)Student\_enrollDB**

CREATE DATABASE stud\_fac;

USE stud\_fac;

CREATE TABLE STUDENT(snum INT PRIMARY KEY,sname VARCHAR(40),major VARCHAR(40),lvl VARCHAR(40),age INT);

CREATE TABLE FACULTY(fid INT PRIMARY KEY,fname VARCHAR(40),deptid INT);

CREATE TABLE CLASS(cname VARCHAR(40) PRIMARY KEY,meets\_at timestamp,room VARCHAR(40),fid INT,FOREIGN KEY(fid) REFERENCES faculty(fid));

CREATE TABLE ENROLLED(snum INT,cname VARCHAR(40),PRIMARY KEY(snum,cname),FOREIGN KEY(snum) REFERENCES STUDENT(snum),FOREIGN KEY(cname) references CLASS(cname));

insert into Student values(1,'jhon', 'CS', 'Sr', 19),

(2, 'Smith', 'CS', 'Jr', 20),

(3 , 'Jacob', 'CV', 'Sr', 20),

(4, 'Tom ', 'CS', 'Jr', 20),

(5, 'Rahul', 'CS', 'Jr', 20),

(6, 'Rita', 'CS', 'Sr', 21);

select \* from Student;

insert into faculty values(11, 'Harish', 1000),

(12, 'MV', 1000),

(13 , 'Mira', 1001),

(14, 'Shiva', 1002),

(15, 'Nupur', 1000);

select \* from Faculty;

insert into Class values('class1', '12/11/15 10:15:16', 'R1', 14),

('class10', '12/11/15 10:15:16', 'R128', 14),

('class2', '12/11/15 10:15:20', 'R2', 12),

('class3', '12/11/15 10:15:25', 'R3', 11),

('class4', '12/11/15 20:15:20', 'R4', 14),

('class5', '12/11/15 20:15:20', 'R3', 15),

('class6', '12/11/15 13:20:20', 'R2', 14),

('class7', '12/11/15 10:10:10', 'R3', 14);

select\*from Class;

insert into Enrolled values

(1,'class1'),

(2,'class1'),

(3,'class3'),

(4,'class3'),

(5,'class4'),

(1,'class5'),

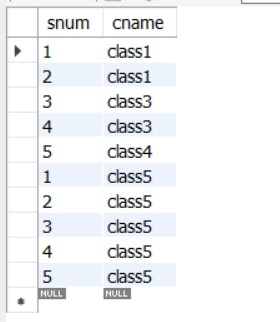
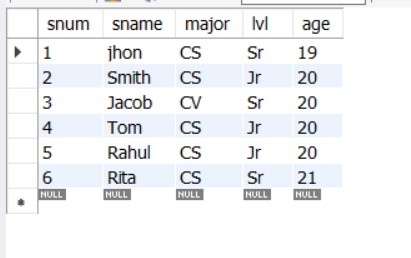
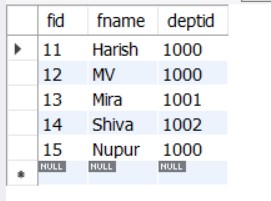
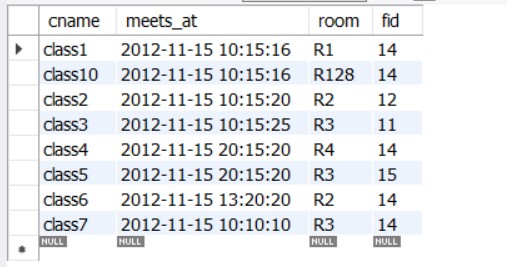
(2,'class5'),

(3,'class5'),

(4,'class5'),

(5,'class5');

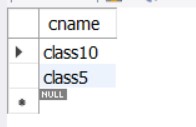
**Tables:-**



**Outputs:-**

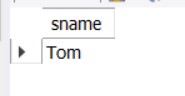
/\*i. Find the names of all Juniors (level = JR) who are enrolled in a class taught by Harish \*/

SELECT sname FROM student,faculty,class,enrolled WHERE student.snum= enrolled.snum and enrolled.cname=class.cname and faculty.fid=class.fid and lvl='Jr' and Faculty.fname='Harish';



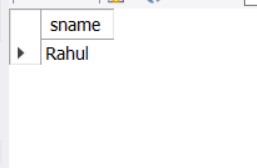
/\*ii. Find the names of all classes that either meet in room R128 or have five or more Students enrolled.\*/

SELECT class.cname FROM class where room='R128' OR class.cname IN(SELECT enrolled.cname FROM enrolled GROUP BY enrolled.cname having COUNT(\*)>=5);



/\*iii. Find the names of all students who are enrolled in two classes that meet at the same time.\*/

select sname from Student where snum in (select e1.snum from Enrolled e1,Enrolled e2,Class c1,Class c2 where e1.snum=e2.snum and e1.cname=c1.cname and e2.cname=c2.cname and e1.cname<>e2.cname and c1.meets\_at=c2.meets\_at);/\*student no is same and meeting time is the same\*/



/\*iv. Find the names of faculty members who teach in every room in which some class is taught.\*/

SELECT fname FROM faculty WHERE NOT EXISTS(select room from class where room not in(select distinct room WHERE faculty.fid=class.fid));/\*innner query selects a room in which they do not teach\*/



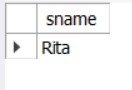
/\*v. Find the names of faculty members for whom the combined enrollment of the courses that they

teach is less than five.\*/

SELECT distinct fname FROM faculty WHERE 5>(SELECT count(enrolled.snum) from enrolled,class where enrolled.cname=class.cname AND class.fid=faculty.fid);

/\*vi. Find the names of students who are not enrolled in any class.\*/

select sname from Student where snum not in (select snum from Enrolled);



/\*vii. For each age value that appears in Students, find the level value that appears most often. For

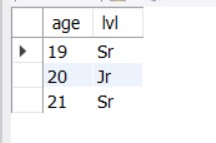
example, if there are more FR level students aged 18 than SR, JR, or SO students aged 18, you

should print the pair (18, FR).\*/

select S.age, S.lvl from Student S group by S.age, S.lvl having S.lvl in (select S1.lvl from Student S1 where S1.age = S.age

group by S1.lvl, S1.age having count(\*) >= all (select count(\*) from Student S2

where s1.age = S2.age group by S2.lvl, S2.age));



**10)SuppliersDB**

create database suppliersdb;

use supplier;

/\*q1\*/

SET FOREIGN\_KEY\_CHECKS=0;

create table suppliers(sid INT PRIMARY KEY,sname VARCHAR(40),address VARCHAR(40));

/\*q2\*/

INSERT INTO suppliers(sid,sname,address) VALUES(10001,'Acme Widget','Bangalore'),

(10002,'Johns','Kolkata'),

(10003,'Vimal','Mumbai'),

(10004,'Reliance','Delhi');

create table parts(pid INT PRIMARY KEY,pname VARCHAR(40),color varchar(40));

INSERT INTO parts(pid,pname,color) VALUES

(20001,'Book','Red'),

(20002,'Pen','Red'),

(20003,'Pencil','Green'),

(20004,'Mobile','Green'),

(20005,'Charger','Black');

create table catalog(sid INT,pid INT,COST real,primary key(sid,pid),FOREIGN KEY(sid) REFERENCES suppliers(sid),FOREIGN KEY(pid) REFERENCES parts(pid));

show tables;

DESC catalog;

DESC parts;

DESC suppliers;

INSERT INTO catalog(sid,pid,COST) VALUES

(10001,20001,10),

(10001,20002,10),

(10001,20003,30),

(10001,20004,10),

(10001,20005,10),

(10002,20001,10),

(10002,20002,20),

(10003,20003,30),

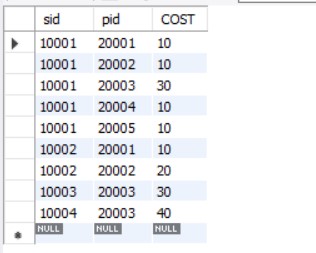
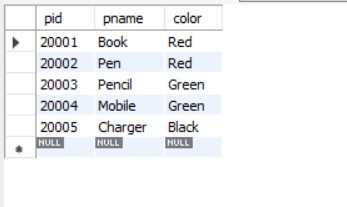
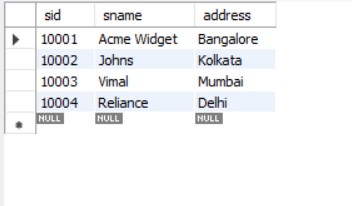
(10004,20003,40);

SELECT\*FROM catalog;

SELECT\*FROM parts;

SELECT\*FROM suppliers;

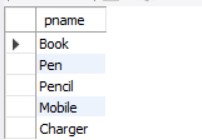
**Tables:-**



**Outputs:-**

/\*1)Find the pnames of parts for which there is some supplier.\*/

SELECT distinct pname FROM parts,catalog WHERE parts.pid=catalog.pid;



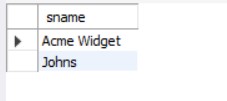
/\*2)Find the snames of suppliers who supply every part.\*/

SELECT sname FROM suppliers,parts,catalog WHERE suppliers.sid=catalog.sid GROUP BY Catalog.sid AND catalog.pid=ALL(SELECT distinct pid FROM parts) ;



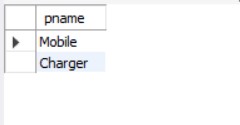
/\*3) Find the snames of suppliers who supply every red part.\*/

SELECT DISTINCT sname FROM suppliers,catalog,parts WHERE suppliers.sid=catalog.sid AND catalog.pid=PARTS.pid AND parts.color='red';



/\*4)Find the pnames of parts supplied by Acme Widget Suppliers and by no one else.\*/

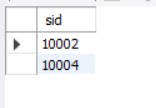
select pname from Parts,Catalog,Suppliers where Catalog.pid=Parts.pid and Catalog.sid=Suppliers.sid and Suppliers.sname='Acme Widget' and Catalog.pid not in (select c.pid from Catalog c ,Suppliers s where s.sid=c.sid and s.sname<>'Acme Widget');/\*not equal <>\*/



/\*5) Find the sids of suppliers who charge more for some part than the average cost of that part (averaged

over all the suppliers who supply that part).\*/

SELECT distinct c.sid FROM catalog c WHERE c.cost>(SELECT AVG(c1.cost) FROM catalog c1 WHERE c1.pid=c.pid);/\*doing self join so that it sums up only costs of dif suppliers\*/



/\*6) For each part, find the sname of the supplier who charges the most for that part.\*/

SELECT p.pid,s.sname from parts p,suppliers s,Catalog c where c.pid=p.pid and c.sid=s.sid and c.cost=(select max(c1.cost) from catalog c1 where c1.pid=p.pid);

