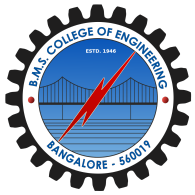


B.M.S. COLLEGE OF ENGINEERING

(AUTONOMOUS COLLEGE UNDER VTU)
BENGALURU-19

DEPARTMENT OF COMPUTER SCIENCE
AND ENGINEERING

DBMS LAB REPORT

(2020-2021)

NAME : VAISHNAVI JAGDALE

USN : 1BM19CS215

COURSE NAME : DATABASE MANAGEMENT SYSTEMS

COURSE TITLE : 19CS4PCDBM

SEM : 4D

PROGRAM 1: Insurance Database

Consider the Insurance database given below. The data types are specified.

PERSON (driver_id: String, name: String, address: String)

CAR (reg_num: String, model: String, year: int)

ACCIDENT (report_num: int, accident_date: date, location: String)

OWNS (driver_id: String, reg_num: String)

PARTICIPATED (driver_id: String,reg_num: String, report_num: int, damage_amount: int)

i) Create the above tables by properly specifying the primary keys and the foreign keys.

create database Insurance2;

use Insurance2;

create table person(driverid varchar(10),name varchar(20), address varchar(30),primary key(driverid));

desc person;

	Field	Type	Null	Key	Default	Extra
▶	driverid	varchar(10)	NO	PRI	NULL	
	name	varchar(20)	YES		NULL	
	address	varchar(30)	YES		NULL	

create table car(regno varchar(10)primary key,model varchar(10),year int);

desc car;

	Field	Type	Null	Key	Default	Extra
▶	regno	varchar(10)	NO	PRI	NULL	
	model	varchar(10)	YES		NULL	
	year	int	YES		NULL	

create table accident(reportno int,adate date,location varchar(20),primary key(reportno));

desc accident;

	Field	Type	Null	Key	Default	Extra
▶	reportno	int	NO	PRI	NULL	
	adate	date	YES		NULL	
	location	varchar(20)	YES		NULL	

create table owns(driverid varchar(10),regno varchar(10), primary key(driverid,regno),foreign key(driverid) references person(driverid),foreign key(regno) references car(regno));

desc owns;

	Field	Type	Null	Key	Default	Extra
▶	driverid	varchar(10)	NO	PRI	NULL	
	regno	varchar(10)	NO	PRI	NULL	

create table participated(driverid varchar(10), regno varchar(10),reportno int, damageamt int, primary key(driverid,regno,reportno), foreign key(driverid) references person(driverid), foreign key(regno) references car(regno), foreign key(reportno) references accident(reportno));

desc participated;

	Field	Type	Null	Key	Default	Extra
▶	driverid	varchar(10)	NO	PRI	NULL	
	regno	varchar(10)	NO	PRI	NULL	
	reportno	int	NO	PRI	NULL	
	damageamt	int	YES		NULL	

ii)Enter at least five tuples for each relation.

insert into person values('A01','Richard','Srinivas Nagar');

insert into person values('A02','Pradeep','Rajajinagar');

insert into person values('A03','Smith','Ashoknagar');

insert into person values('A04','Venu','N.R.Colony');

insert into person values('A05','John','Hanumanth Nagar');

commit;

select * from person;

	driverid	name	address
▶	A01	Richard	Srinivas Nagar
	A02	Pradeep	Rajajinagar
	A03	Smith	Ashoknagar
	A04	Venu	N.R.Colony
	A05	John	Hanumanth Nagar
*	NULL	NULL	NULL

insert into car values('KA052250','Indica', 1990);

insert into car values('KA031181','Lancer', 1957);

insert into car values('KA095477','Toyota', 1998);

insert into car values('KA053408','Honda', 2008);

insert into car values('KA041702','Audi', 2005);

commit;

select * from car;

	regno	model	year
▶	KA031181	Lancer	1957
	KA041702	Audi	2005
	KA052250	Indica	1990
	KA053408	Honda	2008
	KA095477	Toyota	1998
*	NULL	NULL	NULL

insert into accident values(11,'03-01-01','Mysore Road');

insert into accident values(12,'04-02-02','Southend Circle');

insert into accident values(13,'03-01-21','Bulltemple Road');

insert into accident values(14,'08-02-17','Mysore Road');

insert into accident values(15,'05-03-04','Kanakpura Road');

commit;

select * from accident;

	reportno	adate	location
▶	11	2003-01-01	Mysore Road
	12	2004-02-02	Southend Circle
	13	2003-01-21	Bulltemple Road
	14	2008-02-17	Mysore Road
	15	2005-03-04	Kanakpura Road

insert into owns values('A01','KA052250');

insert into owns values('A02','KA053408');

insert into owns values('A03','KA095477');

insert into owns values('A04','KA031181');

insert into owns values('A05','KA041702');

commit;

select * from owns;

	driverid	regno
▶	A04	KA031181
	A05	KA041702
	A01	KA052250
	A02	KA053408
	A03	KA095477
*	NULL	NULL

insert into participated values ('A01','KA052250',11, 10000);

insert into participated values ('A02','KA053408',12, 50000);

insert into participated values ('A03','KA095477',13, 25000);

insert into participated values ('A04','KA031181',14, 3000);

insert into participated values ('A05','KA041702',15, 5000);

commit;

select * from participated;

	driverid	regno	reportno	damageamt
▶	A01	KA052250	11	10000
	A02	KA053408	12	50000
	A03	KA095477	13	25000
	A04	KA031181	14	3000
	A05	KA041702	15	5000
*	NULL	NULL	NULL	NULL

iii)Demonstrate how you

a.Update the damage amount to 25000 for the car with a specific reg-num(example 'K A053408') for which the accident report number was 12.

b.Add a new accident to the database.

update participated set damageamt=25000 where regno='KA053408' and reportno=12;

commit;

select * from participated;

	driverid	regno	reportno	damageamt
▶	A01	KA052250	11	10000
	A02	KA053408	12	25000
	A03	KA095477	13	25000
	A04	KA031181	14	3000
	A05	KA041702	15	5000
*	NULL	NULL	NULL	NULL

insert into accident values(16,'08-03-15','Domlur');

select * from accident;

	reportno	adate	location
▶	11	2003-01-01	Mysore Road
	12	2004-02-02	Southend Circle
	13	2003-01-21	Bulltemple Road
	14	2008-02-17	Mysore Road
	15	2005-03-04	Kanakpura Road
	16	2008-03-15	Domlur
*	NULL	NULL	NULL

iv)Find the total number of people who owned cars that were involved in accidents in 2008.

select count(distinct driverid) CNT from participated a, accident b where a.reportno=b.reportno and year(b.adate)=2008;

	CNT
▶	1

v)Find the number of accidents in which cars belonging to a specific model (example) were involved.

select count(reportno) CNT from car c,participated p where c.regno=p.regno and model='Lancer';

	CNT
▶	1

PROGRAM 2: Banking Enterprise Database

Consider the following database for a banking enterprise.

Branch (branch-name: String, branch-city: String, assets: real)

BankAccount(accno: int, branch-name: String, balance: real)

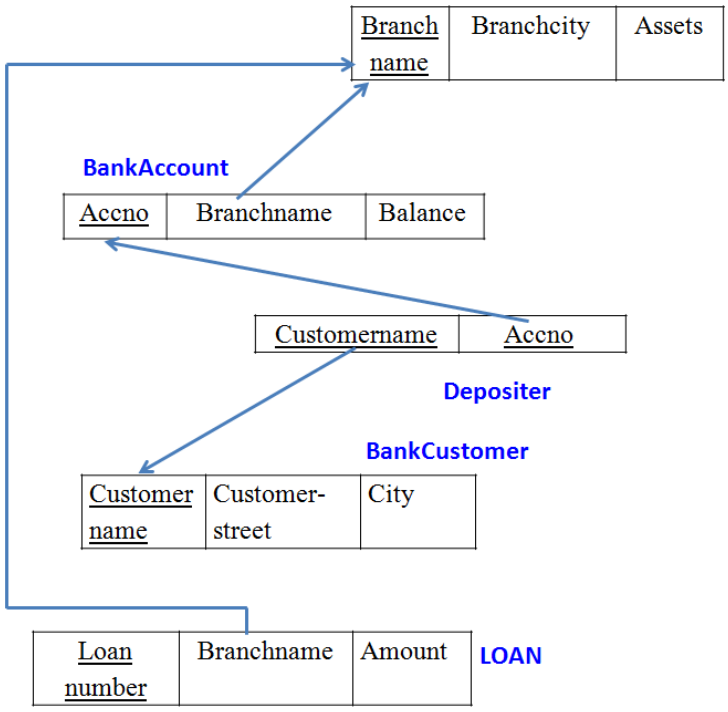
BankCustomer (customer-name: String, customer-street: String, customer-city: String)

Depositer(customer-name: String, accno: int)

Loan (loan-number: int, branch-name: String, amount: real)

INTRODUCTION: This database is developed for supporting banking facilities. Details of the branch along with the accounts and loans handled by them are recorded. Also details of the depositors of the corresponding branches are maintained.

Schema Diagram



i. Create the above tables by properly specifying the primary keys and the foreign keys.

create database bank;

use bank;

create table Branch(branchname varchar(30),branchcity varchar(30),assets real,primary key(branchname));

desc Branch;

Field	Type	Null	Key	Default	Extra
branchname	varchar(30)	NO	PRI	NULL	
branchcity	varchar(30)	YES		NULL	
assets	double	YES		NULL	

create table BankAccount(accno int,branchname varchar(30),balance real,primary key(accno),foreign key(branchname) references Branch(branchname));

desc BankAccount;

Field	Type	Null	Key	Default	Extra
accno	int	NO	PRI	NULL	
branchname	varchar(30)	YES	MUL	NULL	
balance	double	YES		NULL	

create table BankCustomer(customername varchar(30), customerstreet varchar(30), customercity varchar(30), primary key(customername));

desc BankCustomer;

Field	Type	Null	Key	Default	Extra
customername	varchar(30)	NO	PRI	NULL	
customerstreet	varchar(30)	YES		NULL	
customercity	varchar(30)	YES		NULL	

create table Depositer(customername varchar(30),accno integer, primary key(customername,accno),foreign key(customername) references BankCustomer(customername), foreign key(accno) references BankAccount (accno));

desc Depositer;

	Field	Type	Null	Key	Default	Extra
▶	customername	varchar(30)	NO	PRI	NULL	
	accno	int	NO	PRI	NULL	

create table Loan (loannumber int, branchname varchar(30),amount real, primary key(loannumber), foreign key (branchname) references Branch(branchname));

desc Loan;

	Field	Type	Null	Key	Default	Extra
▶	loannumber	int	NO	PRI	NULL	
	branchname	varchar(30)	YES	MUL	NULL	
	amount	double	YES		NULL	

ii. Enter at least five tuples for each relation.

insert into Branch values('SBI_Chamrajpet','Bangalore',50000);

insert into Branch values('SBI_ResidencyRoad','Bangalore',10000);

insert into Branch values('SBI_ShivajiRoad','Bombay',20000);

insert into Branch values('SBI_ParlimentRoad','Delhi',10000);

insert into Branch values('SBI_Jantarmantar','Delhi',20000);

commit;

select * from Branch;

	branchname	branchcity	assets
▶	SBI_Chamrajpet	Bangalore	50000
	SBI_Jantarmantar	Delhi	20000
	SBI_ParlimentRoad	Delhi	10000
	SBI_ResidencyRoad	Bangalore	10000
	SBI_ShivajiRoad	Bombay	20000
*	NULL	NULL	NULL

insert into Loan values(1, 'SBI_Chamrajpet',1000);

insert into Loan values(2, 'SBI_ResidencyRoad',2000);

insert into Loan values(3, 'SBI_ShivajiRoad',3000);

```
insert into Loan values(4, 'SBI_ParlimentRoad',4000);
insert into Loan values(5, 'SBI_Jantarmantar',5000);
commit;
select * from Loan;
```

loannumber	branchname	amount
1	SBI_Chamrajpet	1000
2	SBI_ResidencyRoad	2000
3	SBI_ShivajiRoad	3000
4	SBI_ParlimentRoad	4000
5	SBI_Jantarmantar	5000
NULL	NULL	NULL

```
insert into BankAccount values(1,'SBI_Chamrajpet',2000);
insert into BankAccount values(2,'SBI_ResidencyRoad',5000);
insert into BankAccount values(3,'SBI_ShivajiRoad',6000);
insert into BankAccount values(4,'SBI_ParlimentRoad',9000);
insert into BankAccount values(5,'SBI_Jantarmantar',8000);
insert into BankAccount values(6,'SBI_ShivajiRoad',4000);
insert into BankAccount values(8,'SBI_ResidencyRoad',4000);
insert into BankAccount values(9,'SBI_ParlimentRoad',3000);
insert into BankAccount values(10,'SBI_ResidencyRoad',5000);
insert into BankAccount values(11,'SBI_Jantarmantar',2000);
commit;
select * from BankAccount;
```

accno	branchname	balance
1	SBI_Chamrajpet	2000
2	SBI_ResidencyRoad	5000
3	SBI_ShivajiRoad	6000
4	SBI_ParlimentRoad	9000
5	SBI_Jantarmantar	8000
6	SBI_ShivajiRoad	4000
8	SBI_ResidencyRoad	4000
9	SBI_ParlimentRoad	3000
10	SBI_ResidencyRoad	5000
11	SBI_Jantarmantar	2000
NULL	NULL	NULL

insert into BankCustomer values("Avinash","Bull_Temple_Road","Bangalore");

insert into BankCustomer values("Dinesh","Bannergatta_Road","Bangalore");

insert into BankCustomer values("Mohan","NationalCollege_Road","Bangalore");

insert into BankCustomer values("Nikil","Akbar_Road","Delhi");

insert into BankCustomer values("Ravi","Prithiviraj_Road","Delhi");

commit;

select * from BankCustomer;

customername	customerstreet	customercity
Avinash	Bull_Temple_Road	Bangalore
Dinesh	Bannergatta_Road	Bangalore
Mohan	NationalCollege_Road	Bangalore
Nikil	Akbar_Road	Delhi
Ravi	Prithiviraj_Road	Delhi
NULL	NULL	NULL

insert into Depositer values("Avinash",1);

insert into Depositer values("Dinesh",2);

insert into Depositer values("Nikil",4);

insert into Depositer values("Ravi",5);

insert into Depositer values("Avinash",8);

insert into Depositer values("Nikil",9);

insert into Depositer values("Dinesh",10);

insert into Depositer values("Nikil",11);

commit;

select * from Depositer;

customername	accno
Avinash	1
Dinesh	2
Nikil	4
Ravi	5
Avinash	8
Nikil	9
Dinesh	10
Nikil	11
NULL	NULL

iii. Find all the customers who have at least two accounts at the *Main* branch (ex. SBI_ResidencyRoad).

SELECT c.customername FROM BankCustomer c WHERE EXISTS(SELECT

d.customername,COUNT(d.customername) FROM Depositer d, BankAccount ba WHERE

d.accno=ba.accno AND c.customername=d.customername AND ba.branchname='SBI_ResidencyRoad' GROUP BY d.customername HAVING COUNT(d.customername)>=2);

customername
Dinesh
NULL

iv. Find all the customers who have an account at *all* the branches located in a

specific city (Ex. Delhi).

select distinct d.customername from Depositer d where exists(select * from BankAccount ba

where ba.accno=d.accno and exists (select * from Branch b where b.branchname =

ba.branchname and b.branchcity='Delhi'));

	customername
▶	Ravi
	Nikil

v. Demonstrate how you delete all account tuples at every branch located in

a specific city (Ex. Bombay).

DELETE FROM BankAccount WHERE branchname IN (SELECT branchname FROM BRANCH

WHERE branchcity='Bombay');

select * from BankAccount;

	accno	branchname	balance
▶	1	SBI_Chamrajpet	2000
	2	SBI_ResidencyRoad	5000
	4	SBI_ParlimentRoad	9000
	5	SBI_Jantarmantar	8000
	8	SBI_ResidencyRoad	4000
	9	SBI_ParlimentRoad	3000
	10	SBI_ResidencyRoad	5000
	11	SBI_Jantarmantar	2000
	NULL	NULL	NULL

PROGRAM 3: Supplier Database

Consider the following schema:

SUPPLIERS(sid: integer, sname: string, address: string)

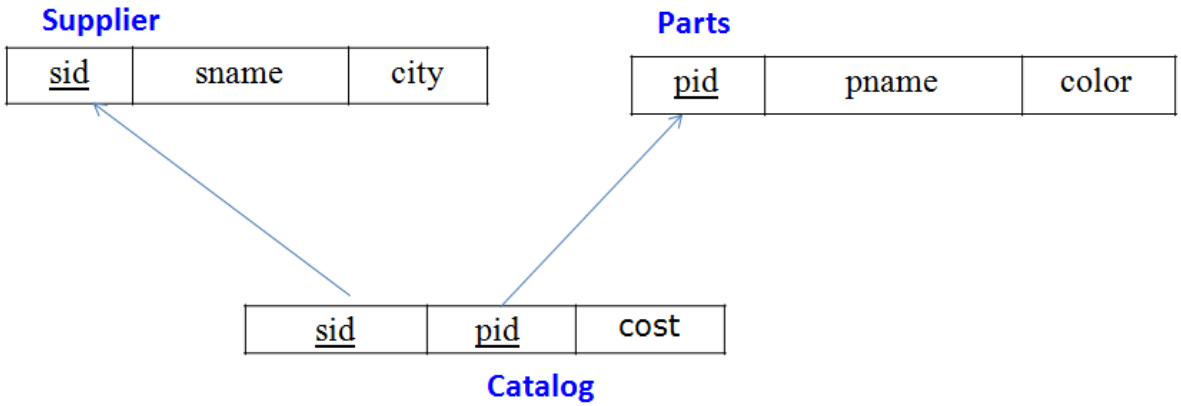
PARTS(pid: integer, pname: string, color: string)

CATALOG(sid: integer, pid: integer, cost: real)

The Catalog relation lists the prices charged for parts by Suppliers.

Write the following queries in SQL:

Schema Diagram




```
create database supplier_database;
use supplier_database;
create table SUPPLIERS(sid int(5) primary key, sname varchar(20), city varchar(20));
desc SUPPLIERS;
```

	Field	Type	Null	Key	Default	Extra
▶	sid	int	NO	PRI	NULL	
	sname	varchar(20)	YES		NULL	
	city	varchar(20)	YES		NULL	

create table PARTS(pid int(5) primary key, pname varchar(20), color varchar(10));

desc PARTS;

	Field	Type	Null	Key	Default	Extra
▶	pid	int	NO	PRI	NULL	
	pname	varchar(20)	YES		NULL	
	color	varchar(10)	YES		NULL	

create table CATALOG(sid int(10), pid int(15), foreign key(sid) references SUPPLIERS(sid), foreign key(pid) references PARTS(pid), cost float(6), primary key(sid, pid));

desc CATALOG;

	Field	Type	Null	Key	Default	Extra
▶	sid	int	NO	PRI	NULL	
	pid	int	NO	PRI	NULL	
	cost	float	YES		NULL	

insert into suppliers values(10001, 'Acme Widget','Bangalore');

insert into suppliers values(10002, 'Johns','Kolkata');

insert into suppliers values(10003, 'Vimal','Mumbai');

insert into suppliers values(10004, 'Reliance','Delhi');

insert into suppliers values(10005, 'Mahindra','Mumbai');

commit;

select * from SUPPLIERS;

	sid	sname	city
▶	10001	Acme Widget	Bangalore
	10002	Johns	Kolkata
	10003	Vimal	Mumbai
	10004	Reliance	Delhi
*	NULL	NULL	NULL

insert into PARTS values(20001, 'Book','Red');
insert into PARTS values(20002, 'Pen','Red');
insert into PARTS values(20003, 'Pencil','Green');
insert into PARTS values(20004, 'Mobile','Green');
insert into PARTS values(20005, 'Charger','Black');
commit;
select * from PARTS;

	pid	pname	color
▶	20001	Book	Red
	20002	Pen	Red
	20003	Pencil	Green
	20004	Mobile	Green
	20005	Charger	Black
*	NULL	NULL	NULL

insert into CATALOG values(10001, '20001','10');
insert into CATALOG values(10001, '20002','10');
insert into CATALOG values(10001, '20003','30');
insert into CATALOG values(10001, '20004','10');
insert into CATALOG values(10001, '20005','10');
insert into CATALOG values(10002, '20001','10');
insert into CATALOG values(10002, '20002','20');
insert into CATALOG values(10003, '20003','30');
insert into CATALOG values(10004, '20003','40');
commit;

select * from CATALOG;

	sid	pid	cost
▶	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
*	NULL	NULL	NULL

i)Find the pnames of parts for which there is some supplier

SELECT DISTINCT P.pname

FROM Parts P, Catalog C

WHERE P.pid = C.pid;

	pname
▶	Book
	Pen
	Pencil
	Mobile
	Charger

ii) Find the snames of suppliers who supply every part.

select s.sname from suppliers s where s.sid in (select c.sid from catalog c group by c.sid having count(distinct (c.pid))=(select count(p.pid) from parts p));

	sname
▶	Acme Widget

iii) Find the snames of suppliers who supply every red part.

select s.sname from suppliers s where s.sid in (select ca.sid from catalog ca,parts p where ca.pid=p.pid and p.color='red' group by ca.sid having count(ca.pid)=(select count(*) from parts p where p.color='red'));

	sname
▶	Acme Widget
	Johns

iv) Find the pnames of parts supplied by Acme Widget Suppliers and by no one else.

select ca.pid from catalog ca where ca.sid= (select s.sid from suppliers s where s.sname ='Acme Widget') having (select count(c.pid) from catalog c where c.pid=ca.pid)=1;

	pname
▶	Mobile
	Charger

v) 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);

	sid
▶	10002
	10004

vi) 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);
```

pid	sname
20001	Acme Widget
20004	Acme Widget
20005	Acme Widget
20001	Johns
20002	Johns
20003	Reliance

PROGRAM 4: Student Faculty Database

Consider the following database for student enrollment for course :

STUDENT(snum: integer, sname: string, major: string, lvl: string, age: integer)

CLASS(cname: string, meets at: time, room: string, fid: integer)

ENROLLED(snum: integer, cname: string)

FACULTY(fid: integer, fname: string, deptid: integer)

The meaning of these relations is straightforward; for example, Enrolled has one record per student-class pair such that the student is enrolled in the class. Level(lvl) is a two character code with 4 different values (example: Junior: JR etc)

Write the following queries in SQL. No duplicates should be printed in any of the answers.

create database student_faculty;

use student_faculty;

create table student(snum INT, sname VARCHAR(10), major VARCHAR(2), lvl VARCHAR(2), age INT, primary key(snum));

desc student;

	Field	Type	Null	Key	Default	Extra
▶	snum	int	NO	PRI	NULL	
	sname	varchar(10)	YES		NULL	
	major	varchar(2)	YES		NULL	
	lvl	varchar(2)	YES		NULL	
	age	int	YES		NULL	

create table faculty(fid INT,fname VARCHAR(20),deptid INT, PRIMARY KEY(fid));

desc faculty;

	Field	Type	Null	Key	Default	Extra
▶	fid	int	NO	PRI	NULL	
	fname	varchar(20)	YES		NULL	
	deptid	int	YES		NULL	

CREATE TABLE class(cname VARCHAR(20), metts_at TIMESTAMP, room VARCHAR(10), fid INT, PRIMARY KEY(cname), FOREIGN KEY(fid) REFERENCES faculty(fid));

desc class;

Field	Type	Null	Key	Default	Extra
cname	varchar(20)	NO	PRI	NULL	
metts_at	timestamp	YES		NULL	
room	varchar(10)	YES		NULL	
fid	int	YES	MUL	NULL	

CREATE TABLE enrolled(snum INT,cname VARCHAR(20),PRIMARY KEY(snum,cname),FOREIGN KEY(snum) REFERENCES student(snum),FOREIGN KEY(cname) REFERENCES class(cname));

desc enrolled;

Field	Type	Null	Key	Default	Extra
snum	int	NO	PRI	NULL	
cname	varchar(20)	NO	PRI	NULL	

insert into student values(1, 'john', 'cs', 'sr', 19);

insert into student values(2, 'smith', 'cs', 'jr', 20);

insert into student values(3 , 'jacob', 'cv', 'sr', 20);

insert into student values(4, 'tom ', 'cs', 'jr', 20);

insert into student values(5, 'rahul', 'cs', 'jr', 20);

insert into student values(6, 'rita', 'cs', 'sr', 21);

commit;

select * from student;

	snum	sname	major	lvl	age
▶	1	John	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
*	NULL	NULL	NULL	NULL	NULL

insert into faculty values(11, 'harish', 1000);
insert into faculty values(12, 'mv', 1000);
insert into faculty values(13 , 'mira', 1001);
insert into faculty values(14, 'shiva', 1002);
insert into faculty values(15, 'nupur', 1000);
commit;
select * from faculty;

	fid	fname	deptid
▶	11	Harish	1000
	12	MV	1000
	13	Mira	1001
	14	Shiva	1002
	15	Nupur	1000
*	NULL	NULL	NULL

insert into class values('class1', '12/11/15 10:15:16', 'R1', 14);
insert into class values('class10', '12/11/15 10:15:16', 'R128', 14);
insert into class values('class2', '12/11/15 10:15:20', 'R2', 12);
insert into class values('class3', '12/11/15 10:15:25', 'R3', 11);
insert into class values('class4', '12/11/15 20:15:20', 'R4', 14);
insert into class values('class5', '12/11/15 20:15:20', 'R3', 15);
insert into class values('class6', '12/11/15 13:20:20', 'R2', 14);
insert into class values('class7', '12/11/15 10:10:10', 'R3', 14);

commit;

select * from class;

	cname	metts_at	room	fid
▶	class1	2012-11-15 10:15:16	R1	14
	class10	2012-11-15 10:15:16	R128	14
	class2	2012-11-15 10:15:20	R2	12
	class3	2012-11-15 10:15:25	R3	11
	class4	2012-11-15 20:15:20	R4	14
	class5	2012-11-15 20:15:20	R3	15
	class6	2012-11-15 13:20:20	R2	14
	class7	2012-11-15 10:10:10	R3	14
*	NULL	NULL	NULL	NULL

insert into enrolled values(1, 'class1');

insert into enrolled values(2, 'class1');

insert into enrolled values(3, 'class3');

insert into enrolled values(4, 'class3');

insert into enrolled values(5, 'class4');

insert into enrolled values(1, 'class5');

insert into enrolled values(2, 'class5');

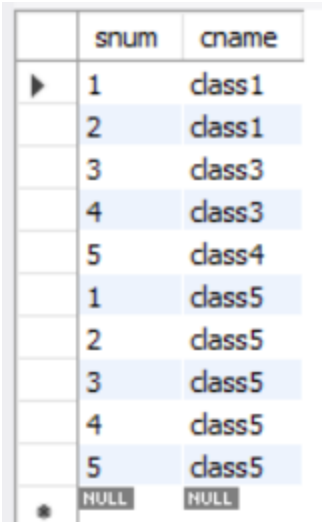
insert into enrolled values(3, 'class5');

insert into enrolled values(4, 'class5');

insert into enrolled values(5, 'class5');

commit;

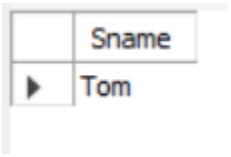
select * from enrolled;



	snum	cname
▶	1	class1
	2	class1
	3	class3
	4	class3
	5	class4
	1	class5
	2	class5
	3	class5
	4	class5
	5	class5
*	NULL	NULL

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

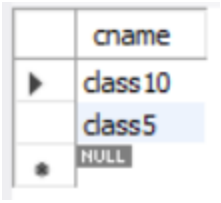
```
SELECT DISTINCT S.Sname FROM Student S, Class C, Enrolled E, Faculty F
WHERE S.snum = E.snum AND E.cname = C.cname AND C.fid = F.fid AND
F.fname = 'Harish' AND S.lvl = 'Jr';
```



	Sname
▶	Tom

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

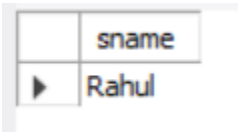
```
SELECT C.cname
FROM class C
WHERE C.room = 'R128'
OR C.cname IN (SELECT E.cname
                FROM enrolled E
                GROUP BY E.cname
                HAVING COUNT(*) >= 5);
```



	cname
▶	class10
	class5
*	NULL

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

```
SELECT DISTINCT S.sname
FROM student S
WHERE S.snum IN (SELECT E1.snum
FROM enrolled E1, enrolled E2, class C1, class C2
WHERE E1.snum = E2.snum AND E1.cname <> E2.cname
AND E1.cname = C1.cname
AND E2.cname = C2.cname AND C1.metts_at = C2.metts_at);
```

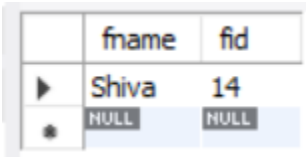


	sname
▶	Rahul

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

```
SELECT f.fname,f.fid FROM faculty f WHERE f.fid in ( SELECT fid FROM class

GROUP BY fid HAVING COUNT(*)=(SELECT COUNT(DISTINCT room) FROM class) );
```



	fname	fid
▶	Shiva	14
*	NULL	NULL

v) Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.

```
SELECT DISTINCT F.fname
FROM faculty F
WHERE 5 > (SELECT COUNT(E.snum)
FROM class C, enrolled E
WHERE C.cname = E.cname
AND C.fid = F.fid);
```

	fname
▶	Harish
	MV
	Mira
	Shiva

vi) Find the names of students who are not enrolled in any class.

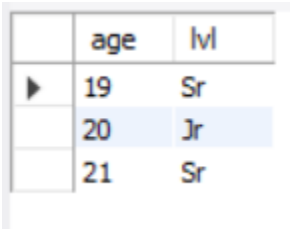
```
SELECT DISTINCT S.sname  FROM Student S
WHERE S.snum NOT IN (SELECT E.snum  FROM Enrolled E );
```

	sname
▶	Rita

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));



	age	lvl
▶	19	Sr
	20	Jr
	21	Sr

PROGRAM 5: Airline Flight Database

Consider the following database that keeps track of airline flight information:

FLIGHTS(flno: integer, from: string, to: string, distance: integer, departs: time, arrives: time, price: integer)

AIRCRAFT(aid: integer, aname: string, cruisingrange: integer)

CERTIFIED(eid: integer, aid: integer)

EMPLOYEES(eid: integer, ename: string, salary: integer)

Note that the Employees relation describes pilots and other kinds of employees as well; Every pilot is certified for some aircraft, and only pilots are certified to fly.

Write each of the following queries in SQL.

```
create database airlines;
 use airlines;
```

```
 create table flights(flno integer primary key, ffrom varchar(15) , tto varchar(15) , distance integer, departs timestamp, arrives timestamp, price integer );
desc flights;
```

	Field	Type	Null	Key	Default	Extra
▶	FLNO	int	NO	PRI	NULL	
	FFROM	varchar(15)	YES		NULL	
	TTO	varchar(15)	YES		NULL	
	DISTANCE	int	YES		NULL	
	DEPARTS	timestamp	YES		NULL	
	ARRIVES	timestamp	YES		NULL	
	PRICE	int	YES		NULL	

```
create table aircraft
  (aid integer primary key,
  aname varchar(10),
  cruisingrange integer);
 desc aircraft;
```

	Field	Type	Null	Key	Default	Extra
▶	AID	int	NO	PRI	NULL	
	ANAME	varchar(10)	YES		NULL	
	CRUISINGRANGE	int	YES		NULL	

```
create table employees
 (eid integer primary key,
 ename varchar(15),
 salary integer );
desc employees;
```

	Field	Type	Null	Key	Default	Extra
▶	EID	int	NO	PRI	NULL	
	ENAME	varchar(15)	YES		NULL	
	SALARY	int	YES		NULL	

```
create table certified
 (eid integer not null,
 aid integer not null,
 primary key (eid, aid),
 foreign key (eid) references employees (eid),
 foreign key (aid) references aircraft (aid));
desc certified;
```

	Field	Type	Null	Key	Default	Extra
▶	EID	int	NO	PRI	NULL	
	AID	int	NO	PRI	NULL	

```
insert into aircraft values(101,'747',3000);
insert into aircraft values(102,'Boeing',900);
insert into aircraft values(103,'647',800);
insert into aircraft values(104,'Dreamliner',10000);
insert into aircraft values(105,'Boeing',3500);
```

insert into aircraft values(106,'707',1500);

insert into aircraft values(107,'Dream', 120000);

commit;

select * from aircraft;

AID	ANAME	CRUISINGRANGE
101	747	3000
102	Boeing	900
103	647	800
104	Dreamliner	10000
105	Boeing	3500
106	707	1500
107	Dream	120000
NULL	NULL	NULL

insert into employees values(701,'A',50000);

insert into employees values(702,'B',100000);

insert into employees values(703,'C',150000);

insert into employees values(704,'D',90000);

insert into employees values(705,'E',40000);

insert into employees values(706,'F',60000);

insert into employees values(707,'G',90000);

select * from employees;

EID	ENAME	SALARY
701	A	50000
702	B	100000
703	C	150000
704	D	90000
705	E	40000
706	F	60000
707	G	90000
NULL	NULL	NULL

insert into certified values(701,101);

```
insert into certified values(701,102);
insert into certified values(701,106);
insert into certified values(701,105);
insert into certified values(702,104);
insert into certified values(703,104);
insert into certified values(704,104);
insert into certified values(702,107);
insert into certified values(703,107);
insert into certified values(704,107);
insert into certified values(702,101);
insert into certified values(703,105);
insert into certified values(704,105);
insert into certified values(705,103);
select * from certified;
```

EID	AID
701	101
702	101
701	102
705	103
702	104
703	104
704	104
701	105
703	105
704	105
701	106
702	107
703	107
704	107

```
insert into flights values(101,'Bangalore','Delhi',2500,TIMESTAMP '2005-05-13 07:15:31',TIMESTAMP '2005-05-13 17:15:31',5000);
```

```
insert into flights values(102,'Bangalore','Lucknow',3000,TIMESTAMP '2005-05-13 07:15:31',TIMESTAMP '2005-05-13 11:15:31',6000);
insert into flights values(103,'Lucknow','Delhi',500,TIMESTAMP '2005-05-13 12:15:31',TIMESTAMP ' 2005-05-13 17:15:31',3000);
insert into flights values(107,'Bangalore','Frankfurt',8000,TIMESTAMP '2005-05-13 07:15:31',TIMESTAMP '2005-05-13 22:15:31',60000);
insert into flights values(104,'Bangalore','Frankfurt',8500,TIMESTAMP '2005-05-13 07:15:31',TIMESTAMP '2005-05-13 23:15:31',75000);
insert into flights values(105,'Kolkata','Delhi',3400,TIMESTAMP '2005-05-13 07:15:31',TIMESTAMP  '2005-05-13 09:15:31',7000);
select * from Flights;
```

FLNO	FFROM	TTO	DISTANCE	DEPARTS	ARRIVES	PRICE
101	Bangalore	Delhi	2500	2005-05-13 07:15:31	2005-05-13 17:15:31	5000
102	Bangalore	Lucknow	3000	2005-05-13 07:15:31	2005-05-13 11:15:31	6000
103	Lucknow	Delhi	500	2005-05-13 12:15:31	2005-05-13 17:15:31	3000
104	Bangalore	Frankfurt	8500	2005-05-13 07:15:31	2005-05-13 23:15:31	75000
105	Kolkata	Delhi	3400	2005-05-13 07:15:31	2005-05-13 09:15:31	7000
107	Bangalore	Frankfurt	8000	2005-05-13 07:15:31	2005-05-13 22:15:31	60000
NULL	NULL	NULL	NULL	NULL	NULL	NULL

i. Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80,000.

```
select distinct a.aname
from aircraft a
where a.aid in (select c.aid
 from certified c, employees e
 where c.eid = e.eid and
 not exists ( select *
 from employees e1
 where e1.eid = e.eid and e1.salary <80000 ));
```

	aname
▶	747
	Dreamliner
	Boeing
	Dream

ii. For each pilot who is certified for more than three aircrafts, find the eid and the maximum cruisingrange of the aircraft for which she or he is certified.

```
select c.eid, max(a.cruisingrange)
from certified c, aircraft a
where c.aid = a.aid
group by c.eid
having count(*) > 3;
```

	eid	MAX(A.cruisingrange)
▶	701	3500

iii. Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Frankfurt.

```
SELECT DISTINCT E.ename

FROM Employees E
WHERE E.salary <( SELECT MIN(F.price)
                  FROM Flights F
                  WHERE F.ffrom = 'Bangalore' AND F.tto = 'Frankfurt' );
```

	ename
▶	A
	E

iv. For all aircraft with cruisingrange over 1000 Kms, find the name of the aircraft and the average salary of all pilots certified for this aircraft.

```
SELECT Temp.name, Temp.AvgSalary

FROM ( SELECT A.aid, A.aname AS name, AVG (E.salary)  AS AvgSalary
 FROM Aircraft A, Certified C, Employees E
```

```
WHERE A.aid = C.aid AND C.eid = E.eid AND A.cruisingrange > 1000
GROUP BY A.aid, A.aname )  Temp;
```

name	AvgSalary
747	75000.0000
Dreamliner	113333.3333
Boeing	96666.6667
707	50000.0000
Dream	113333.3333

v. Find the names of pilots certified for some Boeing aircraft.

```
SELECT DISTINCT E.ename
FROM Employees E, Certified C, Aircraft A
WHERE E.eid = C.eid AND C.aid = A.aid AND A.aname  LIKE 'Boeing%';
```

ename
A
C
D

vi. Find the aids of all aircraft that can be used on routes from Bengaluru to New Delhi.

```
SELECT A.aid
FROM Aircraft A
WHERE A.cruisingrange >( SELECT MIN(F.distance)
                         FROM Flights F
                         WHERE F.ffrom = 'Bangalore' AND F.tto = 'Frankfurt' );
```

aid
104
107

vii. A customer wants to travel from Madison to New York with no more than two changes of flight. List the choice of departure times from Madison if the customer wants to arrive in New York by 6 p.m.

```
SELECT F.departs
FROM Flights F
WHERE F.flno IN ( ( SELECT F0.flno
 FROM Flights F0
 WHERE F0.ffrom = 'Bangalore' AND F0.tto = 'Delhi'
 AND extract(hour from F0.arrives) < 18 )
 UNION
( SELECT F0.flno
 FROM Flights F0, Flights F1
 WHERE F0.ffrom = 'Bangalore' AND F0.tto <> 'Delhi'
 AND F0.tto = F1.ffrom AND F1.tto = 'Delhi'
 AND F1.departs > F0.arrives
 AND extract(hour from F1.arrives) < 18)
 UNION
( SELECT F0.flno
 FROM Flights F0, Flights F1, Flights F2
 WHERE F0.ffrom = 'Bangalore'
 AND F0.tto = F1.ffrom
 AND F1.tto = F2.ffrom
 AND F2.tto = 'Delhi'
 AND F0.tto <> 'Delhi'
 AND F1.tto <> 'Delhi'
 AND F1.departs > F0.arrives
 AND F2.departs > F1.arrives
 AND extract(hour from F2.arrives) < 18));
```

	departs
▶	2005-05-13 07:15:31
	2005-05-13 07:15:31

viii. Print the name and salary of every non-pilot whose salary is more than the average salary for pilots.

```
SELECT E.ename, E.salary
FROM Employees E
WHERE E.eid NOT IN ( SELECT DISTINCT C.eid
 FROM Certified C )
AND E.salary >( SELECT AVG (E1.salary)
 FROM Employees E1
 WHERE E1.eid IN
( SELECT DISTINCT C1.eid
 FROM Certified C1 ) );
```

	ename	salary
▶	G	90000

PROGRAM 6: Order Database

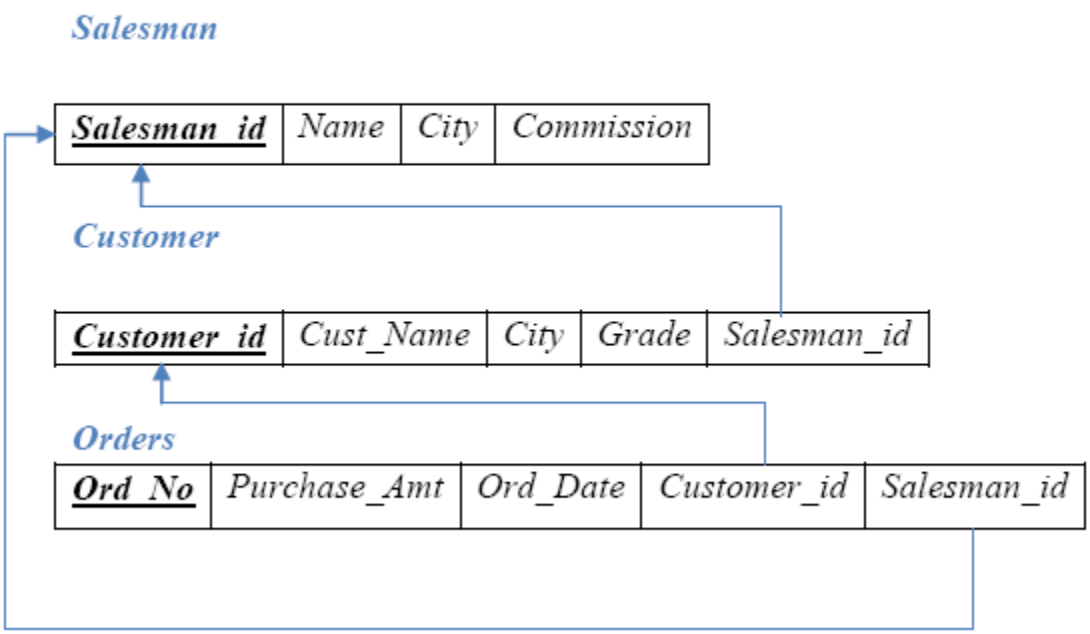
SALESMAN (Salesman_id, Name, City, Commission)

CUSTOMER (Customer_id, Cust_Name, City, Grade, Salesman_id)

ORDERS (Ord_No, Purchase_Amt, Ord_Date, Customer_id, Salesman_id)

Write SQL queries to

Schema Diagram




```
create database Program6;
use Program6;

create table salesman ( salesman_id int,
name varchar (20),
city varchar (20), commission varchar (20), primary key (salesman_id));
desc salesman;
```

	Field	Type	Null	Key	Default	Extra
▶	salesman_id	int	NO	PRI	NULL	
	name	varchar(20)	YES		NULL	
	city	varchar(20)	YES		NULL	
	commission	varchar(20)	YES		NULL	

create table customer (
customer_id int, cust_name varchar (20), city varchar (20),
grade int ,

salesman_id int,
primary key (customer_id),
foreign key (salesman_id) references salesman(salesman_id) on delete set

null);
desc customer;

	Field	Type	Null	Key	Default	Extra
▶	customer_id	int	NO	PRI	NULL	
	cust_name	varchar(20)	YES		NULL	
	city	varchar(20)	YES		NULL	
	grade	int	YES		NULL	
	salesman_id	int	YES	MUL	NULL	

create table orders (ord_no int ,
purchase_amt int,
ord_date date, customer_id int,
salesman_id int,
primary key (ord_no),
foreign key (customer_id) references customer (customer_id) on delete
cascade,
foreign key (salesman_id) references salesman (salesman_id) on delete
cascade);
desc orders;

	Field	Type	Null	Key	Default	Extra
▶	ord_no	int	NO	PRI	NULL	
	purchase_amt	int	YES		NULL	
	ord_date	date	YES		NULL	
	customer_id	int	YES	MUL	NULL	
	salesman_id	int	YES	MUL	NULL	

```
insert into salesman values (1000, 'john','bangalore','25 %');
 insert into salesman values (2000, 'ravi','bangalore','20 %');
 insert into salesman values (3000, 'kumar','mysore','15 %');
insert into salesman values (4000, 'smith','delhi','30 %');
insert into salesman values (5000, 'harsha','hydrabad','15 %');
select * from salesman;
```

salesman_id	name	city	commission
1000	john	bangalore	25 %
2000	ravi	bangalore	20 %
3000	kumar	mysore	15 %
4000	smith	delhi	30 %
5000	harsha	hydrabad	15 %
NULL	NULL	NULL	NULL

```
insert into customer values (10, 'preethi','bangalore', 100, 1000);
 insert into customer values (11,'vivek','mangalore', 300, 1000);
insert into customer values (12, 'bhaskar','chennai', 400, 2000);
insert into customer values (13, 'chethan','bangalore', 200, 2000);
 insert into customer values (14, 'mamatha','bangalore', 400, 3000);
select * from customer;
```

customer_id	cust_name	city	grade	salesman_id
10	preethi	bangalore	100	1000
11	vivek	mangalore	300	1000
12	bhaskar	chennai	400	2000
13	chethan	bangalore	200	2000
14	mamatha	bangalore	400	3000
NULL	NULL	NULL	NULL	NULL

```
insert into orders values (50, 5000, '04-06-17', 10, 1000);
insert into orders values (51, 450, '20-01-17', 10, 2000);
insert into orders values (52, 1000, '24-02-17', 13, 2000);
 insert into orders values (53, 3500, '13-04-17', 14, 3000);
 insert into orders values (54, 550, '09-03-17', 12, 2000);
 select * from orders;
```

	ord_no	purchase_amt	ord_date	customer_id	salesman_id
▶	50	5000	2004-06-17	10	1000
	51	450	2020-01-17	10	2000
	52	1000	2024-02-17	13	2000
	53	3500	2013-04-17	14	3000
	54	550	2009-03-17	12	2000
*	NULL	NULL	NULL	NULL	NULL

i. Count the customers with grades above Bangalore's average.

```
SELECT grade, count(DISTINCT customer_id) FROM customer
GROUP BY grade
HAVING grade > (SELECT AVG(grade)
FROM customer
WHERE city='bangalore');
```

	grade	count(DISTINCT customer_id)
▶	300	1
	400	2

ii. Find the name and numbers of all salesmen who had more than one customer.

```
SELECT salesman_id, NAME FROM salesman a
WHERE 1 < (SELECT count(*) FROM customer
WHERE salesman_id=a.salesman_id);
```

	salesman_id	NAME
▶	1000	john
	2000	ravi
*	NULL	NULL

iii. List all salesmen and indicate those who have and don't have customers in their cities (Use UNION operation.)

```
SELECT salesman.salesman_id, NAME, cust_name, commission FROM salesman, customer
WHERE salesman.city = customer.city
UNION
SELECT salesman_id, name, 'no customer', commission FROM salesman
WHERE NOT city = ANY
(SELECT city
FROM customer) ORDER BY 2 DESC;
```

salesman_id	NAME	cust_name	commission
4000	smith	no customer	30 %
2000	ravi	preethi	20 %
2000	ravi	chethan	20 %
2000	ravi	mamatha	20 %
3000	kumar	no customer	15 %
1000	john	preethi	25 %
1000	john	chethan	25 %
1000	john	mamatha	25 %
5000	harsha	no customer	15 %

iv. Create a view that finds the salesman who has the customer with the highest order of a day.

```
CREATE VIEW highsalesman AS
SELECT b.ord_date, a.salesman_id, a.name
FROM salesman a, orders b
WHERE a.salesman_id = b.salesman_id
AND b.purchase_amt=(SELECT max(purchase_amt) FROM orders c
WHERE c.ord_date = b.ord_date);
SELECT * FROM highsalesman;
```

ord_date	salesman_id	name
2004-06-17	1000	john
2020-01-17	2000	ravi
2024-02-17	2000	ravi
2013-04-17	3000	kumar
2009-03-17	2000	ravi

v. Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.

DELETE FROM salesman WHERE salesman_id=1000; SELECT * FROM salesman; SELECT * FROM orders;

salesman_id	name	city	commission
2000	ravi	bangalore	20 %
3000	kumar	mysore	15 %
4000	smith	delhi	30 %
5000	harsha	hydrabad	15 %
NULL	NULL	NULL	NULL

ord_no	purchase_amt	ord_date	customer_id	salesman_id
51	450	2020-01-17	10	2000
52	1000	2024-02-17	13	2000
53	3500	2013-04-17	14	3000
54	550	2009-03-17	12	2000
NULL	NULL	NULL	NULL	NULL

PROGRAM 7: Books Database

Consider the following database

BOOK (Book_id, Title, Publisher_Name, Pub_Year)

BOOK_AUTHORS (Book_id, Author_Name)

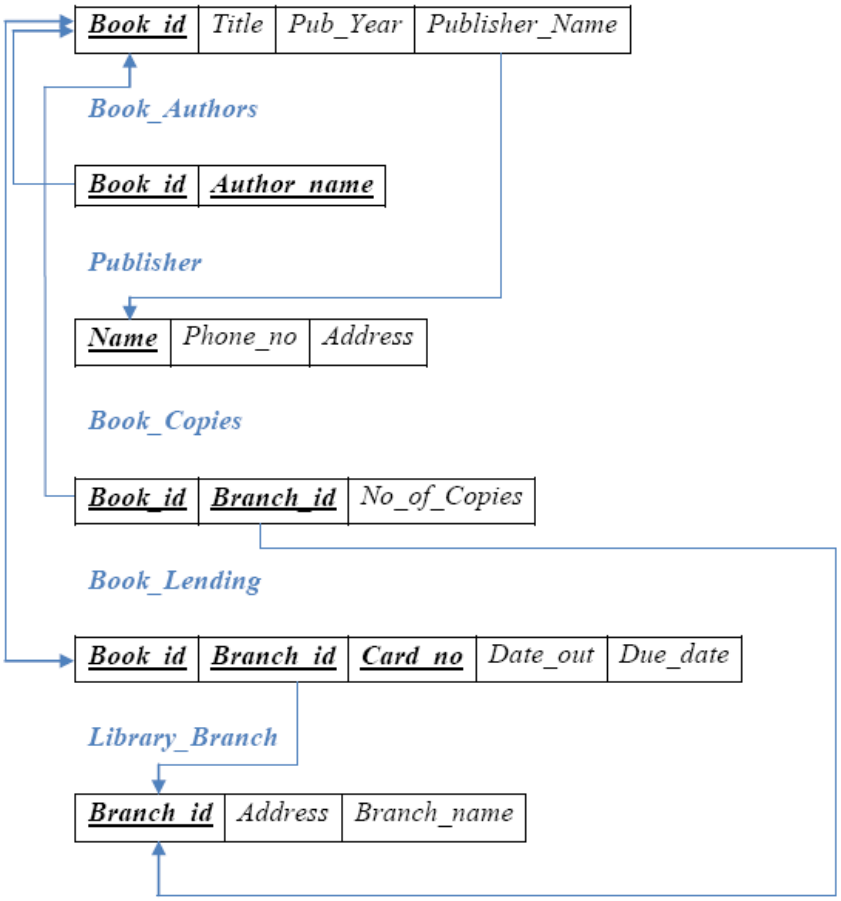
PUBLISHER (Name, Address, Phone)

BOOK_COPIES (Book_id, Branch_id, No-of_Copies)

BOOK_LENDING (Book_id, Branch_id, Card_No, Date_Out, Due_Date)

LIBRARY_BRANCH (Branch_id, Branch_Name, Address)

Schema Diagram




```
create database Lab7;
use Lab7;

create table publisher (
name varchar (20) primary key,
phone integer,
address varchar (20) );
desc publisher;
```

Field	Type	Null	Key	Default	Extra
name	varchar(20)	NO	PRI	NULL	
phone	int	YES		NULL	
address	varchar(20)	YES		NULL	

```
create table book (
book_id integer primary key,
title varchar (20),
pub_year varchar (20),
publisher_name varchar (20),
foreign key (publisher_name) references publisher (name) on delete cascade);
desc book;
```

Field	Type	Null	Key	Default	Extra
book_id	int	NO	PRI	NULL	
title	varchar(20)	YES		NULL	
pub_year	varchar(20)	YES		NULL	
publisher_name	varchar(20)	YES	MUL	NULL	

```
create table book_authors ( author_name varchar (20),
book_id integer,
foreign key (book_id) references book (book_id) on delete cascade, primary key (book_id,
author_name) );
desc book_authors;
```

Field	Type	Null	Key	Default	Extra
author_name	varchar(20)	NO	PRI	NULL	
book_id	int	NO	PRI	NULL	

```
create table library_branch ( branch_id integer primary key,
branch_name varchar (50),
address varchar (50) );
desc library_branch;
```

Field	Type	Null	Key	Default	Extra
branch_id	int	NO	PRI	NULL	
branch_name	varchar(50)	YES		NULL	
address	varchar(50)	YES		NULL	

```
create table book_copies ( no_of_copies integer,
book_id integer,
branch_id integer,
foreign key (book_id) references book (book_id) on delete cascade,
```

foreign key (branch_id) references library_branch (branch_id) on delete cascade, primary key (book_id, branch_id));
desc book_copies;

no_of_copies	book_id	branch_id
10	1	10
5	1	11
2	2	12
5	2	13
7	3	14
3	4	11
1	5	10
NULL	NULL	NULL

create table card (
card_no integer primary key);
desc card;

Field	Type	Null	Key	Default	Extra
card_no	int	NO	PRI	NULL	

create table book_lending (date_out date,
due_date date, book_id integer,
branch_id integer, card_no integer,
foreign key (book_id) references book (book_id) on delete cascade,
foreign key (branch_id) references library_branch (branch_id) on delete cascade, foreign key (card_no) references card (card_no) on delete cascade,
primary key (book_id, branch_id, card_no)
);
desc book_lending;

Field	Type	Null	Key	Default	Extra
date_out	date	YES		NULL	
due_date	date	YES		NULL	
book_id	int	NO	PRI	NULL	
branch_id	int	NO	PRI	NULL	
card_no	int	NO	PRI	NULL	

insert into publisher values ('mcgraw-hill', 99890, 'bangalore'); insert into publisher values ('pearson', 98890, 'newdelhi');
insert into publisher values ('random house', 74556, 'hyderabad'); insert into publisher values ('hachette livre', 897086, 'chennai');
insert into publisher values ('grupo planeta', 77561, 'bangalore');
select * from publisher;

name	phone	address
grupo planeta	77561	bangalore
hachette livre	897086	chenai
mcgraw-hill	99890	bangalore
pearson	98890	newdelhi
random house	74556	hyderabad
NULL	NULL	NULL

insert into book values (1,'dbms','01-2017', 'mcgraw-hill'); insert into book values (2,'adbms','06-2016', 'mcgraw-hill'); insert into book values (3,'cn','09-2016', 'pearson');
insert into book values (4,'cg','09-2015', 'grupo planeta'); insert into book values (5,'os','05-2016', 'pearson');
select * from book;

book_id	title	pub_year	publisher_name
1	dbms	01-2017	mcgraw-hill
2	adbms	06-2016	mcgraw-hill
3	cn	09-2016	pearson
4	cg	09-2015	grupo planeta
5	os	05-2016	pearson
NULL	NULL	NULL	NULL

insert into book_authors values ('navathe', 1); insert into book_authors values ('navathe', 2); insert into book_authors values ('tanenbaum', 3); insert into book_authors values ('edward angel', 4); insert into book_authors values ('galvin', 5);

select * from book_authors;

	author_name	book_id
▶	navathe	1
	navathe	2
	tanenbaum	3
	edward angel	4
	galvin	5
*	NULL	NULL

insert into library_branch values (10,'rr nagar','bangalore'); insert into library_branch values (11,'rnsit','bangalore');
insert into library_branch values (12,'rajaji nagar', 'bangalore'); insert into library_branch values (13,'nitte','mangalore');
insert into library_branch values (14,'manipal','udupi');
select * from library_branch;

	branch_id	branch_name	address
▶	10	rr nagar	bangalore
	11	rnsit	bangalore
	12	rajaji nagar	bangalore
	13	nitte	mangalore
	14	manipal	udupi
*	NULL	NULL	NULL

insert into book_copies values (10, 1, 10); insert into book_copies values (5, 1, 11); insert into book_copies values (2, 2, 12); insert into book_copies values (5, 2, 13); insert into book_copies values (7, 3, 14); insert into book_copies values (1, 5, 10); insert into book_copies values (3, 4, 11);
select * from book_copies;

	Field	Type	Null	Key	Default	Extra
▶	no_of_copies	int	YES		NULL	
	book_id	int	NO	PRI	NULL	
	branch_id	int	NO	PRI	NULL	

insert into card values (100);
insert into card values (101);
insert into card values (102);
insert into card values (103);

```
insert into card values (104);
select * from card;
```

	card_no
▶	100
	101
	102
	103
	104
*	NULL

```
insert into book_lending values ('01-01-17','01-06-17', 1, 10, 101);
insert into book_lending values ('11-01-17','11-03-17', 3, 14, 101);
insert into book_lending values ('21-02-17','21-04-17', 2, 13, 101);
insert into book_lending values ('15-03-17','15-07-17', 4, 11, 101);
insert into book_lending values ('12-08-17','12-08-17', 1, 11, 104);
select * from book_lending;
```

	date_out	due_date	book_id	branch_id	card_no
▶	2001-01-17	2001-06-17	1	10	101
	2012-08-17	2012-08-17	1	11	104
	2021-02-17	2021-04-17	2	13	101
	2011-01-17	2011-03-17	3	14	101
	2015-03-17	2015-07-17	4	11	101
*	NULL	NULL	NULL	NULL	NULL

Write SQL queries to

i.Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each branch, etc.

```
select b.book_id, b.title, b.pub_year, b.publisher_name, bc.no_of_copies, ba.author_name,
lb.branch_name from book b, book_authors ba,
library_branch lb, book_copies bc where b.book_id = ba.book_id and b.book_id = bc.book_id and
lb.branch_id = bc.branch_id;
```

book_id	title	pub_year	publisher_name	no_of_copies	author_name	branch_name
1	dbms	01-2017	mcgraw-hill	10	navathe	rr nagar
1	dbms	01-2017	mcgraw-hill	5	navathe	rnsit
2	adbms	06-2016	mcgraw-hill	2	navathe	rajaji nagar
2	adbms	06-2016	mcgraw-hill	5	navathe	nitte
3	cn	09-2016	pearson	7	tanenbaum	manipal
4	cg	09-2015	grupo planeta	3	edward angel	rnsit
5	os	05-2016	pearson	1	galvin	rr nagar

ii.Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017

select card_no from book_lending where year(date_out) >17 and month(date_out)<7 group by card_no having count(card_no) >2 ;

card_no
101

3.Delete a book in the BOOK table. Update the contents of other tables to reflect this data manipulation operation.

delete from book where book_id = 3; select * from book;
select * from book_authors;
select * from book_copies;
select * from book_lending;

book_id	title	pub_year	publisher_name
1	dbms	01-2017	mcgraw-hill
2	adbms	06-2016	mcgraw-hill
4	cg	09-2015	grupo planeta
5	os	05-2016	pearson
NULL	NULL	NULL	NULL

author_name	book_id
navathe	1
navathe	2
edward angel	4
galvin	5
NULL	NULL

	no_of_copies	book_id	branch_id
▶	10	1	10
	5	1	11
	2	2	12
	5	2	13
	3	4	11
	1	5	10
*	NULL	NULL	NULL

	date_out	due_date	book_id	branch_id	card_no
▶	2001-01-17	2001-06-17	1	10	101
	2012-08-17	2012-08-17	1	11	104
	2021-02-17	2021-04-17	2	13	101
	2015-03-17	2015-07-17	4	11	101
*	NULL	NULL	NULL	NULL	NULL

4.Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.

create view q4_view as select pub_year from book; select * from q4_view;

	pub_year
▶	01-2017
	06-2016
	09-2015
	05-2016

5.Create a view of all books and their number of copies that are currently available in the Library.

create view q5_view as select b.book_id, b.title, bc.no_of_copies from book b, book_copies bc where b.book_id = bc.book_id;
select * from q5_view;

	book_id	title	no_of_copies
▶	1	dbms	10
	1	dbms	5
	2	adbms	2
	2	adbms	5
	4	cg	3
	5	os	1

PROGRAM 8: Student Enrollment

Consider the following database of student enrollment in courses & books adopted for each course.

STUDENT (regno: string, name: string, major: string, bdate:date)

COURSE (course #:int, cname:string, dept:string)

ENROLL (regno:string, course#:int, sem:int, marks:int)

BOOK _ ADOPTION (course# :int, sem:int, book-ISBN:int)

TEXT (book-ISBN:int, book-title:string, publisher:string, author:string)

i. Create the above tables by properly specifying the primary keys and the foreign keys.

```
create database Lab8;
use Lab8;

create table student(
    regno varchar(15),
    name varchar(20),
    major varchar(20),
    bdate date,
    primary key (regno)
    );
desc student;
```

	regno	name	major	bdate
▶	1pe11cs002	b	sr	1993-09-24
	1pe11cs003	c	sr	1993-11-27
	1pe11cs004	d	sr	1993-04-13
	1pe11cs005	e	jr	1994-08-24
*	NULL	NULL	NULL	NULL

```
create table course(
    courseno int,
    cname varchar(20),
    dept varchar(20),
    primary key (courseno)
    );
desc course;
```

	Field	Type	Null	Key	Default	Extra
▶	courseno	int	NO	PRI	NULL	
	cname	varchar(20)	YES		NULL	
	dept	varchar(20)	YES		NULL	

```
create table enroll(
    regno varchar(15),
    courseno int,
    sem int,
    marks int,
    primary key (regno,courseno),
    foreign key (regno) references student (regno),
    foreign key (courseno) references course (courseno)
    );
desc enroll;
```

	Field	Type	Null	Key	Default	Extra
▶	regno	varchar(15)	NO	PRI	NULL	
	courseno	int	NO	PRI	NULL	
	sem	int	YES		NULL	
	marks	int	YES		NULL	

```
create table text(
    book_isbn int,
    book_title varchar(20),
    publisher varchar(20),
    author varchar(20),
    primary key (book_isbn)
    );
desc text;
```

	Field	Type	Null	Key	Default	Extra
▶	book_isbn	int	NO	PRI	NULL	
	book_title	varchar(20)	YES		NULL	
	publisher	varchar(20)	YES		NULL	
	author	varchar(20)	YES		NULL	

```
create table book_adoption(
    courseno int,
    sem int,
    book_isbn int,
    primary key (courseno,book_isbn),
    foreign key (courseno) references course (courseno),
    foreign key (book_isbn) references text(book_isbn)
    );
```

desc book_adoption;

Field	Type	Null	Key	Default	Extra
courseno	int	NO	PRI	NULL	
sem	int	YES		NULL	
book_isbn	int	NO	PRI	NULL	

ii. Enter at least five tuples for each relation.

```
insert into student (regno,name,major,bdate) values
    ('1pe11cs002','b','sr','19930924'),
    ('1pe11cs003','c','sr','19931127'),
    ('1pe11cs004','d','sr','19930413'),
    ('1pe11cs005','e','jr','19940824');
select * from student;
```

regno	name	major	bdate
1pe11cs002	b	sr	1993-09-24
1pe11cs003	c	sr	1993-11-27
1pe11cs004	d	sr	1993-04-13
1pe11cs005	e	jr	1994-08-24
NULL	NULL	NULL	NULL

```
insert into course values (111,'os','cse'),
    (112,'ec','cse'),
    (113,'ss','ise'),
    (114,'dbms','cse'),
    (115,'signals','ece');
select * from course;
```

courseno	cname	dept
111	os	cse
112	ec	cse
113	ss	ise
114	dbms	cse
115	signals	ece
NULL	NULL	NULL

```
insert into text values (book_isbn,book_title,publisher,author),
    (10,'database systems','pearson','schield'),
    (900,'operating sys','pearson','leland'),
    (901,'circuits','hall india','bob'),
    (902,'system software','peterson','jacob'),
    (903,'scheduling','pearson','patil'),
```

```
    (904,'database systems','pearson','jacob'),
    (905,'database manager','pearson','bob'),
    (906,'signals','hall india','sumit');
select * from text;
```

book_isbn	book_title	publisher	author
0	NULL	NULL	NULL
10	database systems	pearson	schield
900	operating sys	pearson	leland
901	circuits	hall india	bob
902	system software	peterson	jacob
903	scheduling	pearson	patil
904	database systems	pearson	jacob
905	database manager	pearson	bob
906	signals	hall india	sumit
NULL	NULL	NULL	NULL

```
insert into enroll (regno,courseno,sem,marks) values
    ('1pe11cs002',114,5,100),
    ('1pe11cs003',113,5,100),
    ('1pe11cs004',111,5,100),
    ('1pe11cs005',112,3,100);
select * from enroll;
```

regno	courseno	sem	marks
1pe11cs002	114	5	100
1pe11cs003	113	5	100
1pe11cs004	111	5	100
1pe11cs005	112	3	100
NULL	NULL	NULL	NULL

```
insert into book_adoption (courseno,sem,book_isbn) values
(111,5,900),
(111,5,903),
(111,5,904),
(112,3,901),
(113,3,10),
(114,5,905),
(113,5,902),
(115,3,906);
select * from book_adoption;
```

courseno	sem	book_isbn
111	5	900
111	5	903
111	5	904
112	3	901
113	3	10
113	5	902
114	5	905
115	3	906
NULL	NULL	NULL

iii. Demonstrate how you add a new text book to the database and make this book be adopted by some department.

insert into text values (907,'ai','hall india','sumit');
insert into book_adoption values(115, 2, 907);
select * from text;
select * from book_adoption;

book_isbn	book_title	publisher	author
0	NULL	NULL	NULL
10	database systems	pearson	schield
900	operating sys	pearson	leland
901	circuits	hall india	bob
902	system software	peterson	jacob
903	scheduling	pearson	patil
904	database systems	pearson	jacob
905	database manager	pearson	bob
906	signals	hall india	sumit
907	ai	hall india	sumit
NULL	NULL	NULL	NULL

courseno	sem	book_isbn
111	5	900
111	5	903
111	5	904
112	3	901
113	3	10
113	5	902
114	5	905
115	3	906
115	2	907
NULL	NULL	NULL

iv. Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses offered by the 'CS' department that use more than two books.

```
select b.book_isbn, b.courseno, t.book_title from book_adoption b, text t where t.book_isbn =
b.book_isbn and b.courseno in (
select courseno from course where dept = 'cse' and courseno in (select courseno from book_adoption
group by courseno having count(*)>2));
```

book_isbn	courseno	book_title
900	111	operating sys
903	111	scheduling
904	111	database systems

v. List any department that has all its adopted books published by a specific publisher.

```
select distinct c.dept
    from course c
    where c.dept in
    ( select c.dept
    from course c,book_adoption b,text t
    where c.courseno=b.courseno
    and t.book_isbn=b.book_isbn
    and t.publisher='hall india')
    and c.dept not in
    (select c.dept
    from course c,book_adoption b,text t
    where c.courseno=b.courseno
    and t.book_isbn=b.book_isbn
    and t.publisher != 'hall india');
```

dept
ece

PROGRAM 9: Movie Database

Consider the schema for Movie Database:

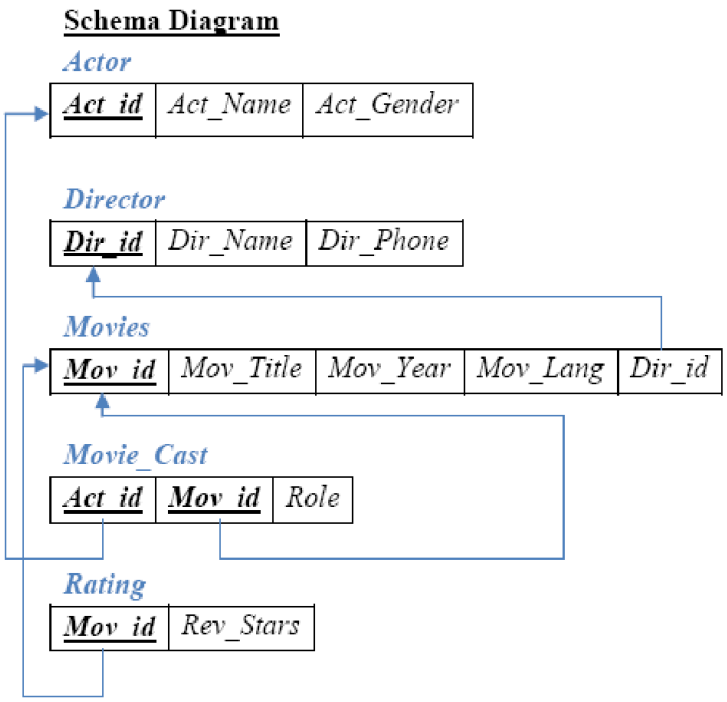
ACTOR (Act_id, Act_Name, Act_Gender)

DIRECTOR (Dir_id, Dir_Name, Dir_Phone)

MOVIES (Mov_id, Mov_Title, Mov_Year, Mov_Lang, Dir_id)

MOVIE_CAST (Act_id, Mov_id, Role)

RATING (Mov_id, Rev_Stars)




```
create database Lab9;
use Lab9;
create table actor(act_id int, act_name varchar(30), act_gender enum('M','F'), primary key(act_id));
desc actor;
```

Field	Type	Null	Key	Default	Extra
act_id	int	NO	PRI	NULL	
act_name	varchar(30)	YES		NULL	
act_gender	enum('M','F')	YES		NULL	

create table director(dir_id int, dir_name varchar(30), dir_phone varchar(10), primary key(dir_id));
desc director;

Field	Type	Null	Key	Default	Extra
dir_id	int	NO	PRI	NULL	
dir_name	varchar(30)	YES		NULL	
dir_phone	varchar(10)	YES		NULL	

create table movies(mov_id int, mov_title varchar(30), mov_year year, mov_lang varchar(10), dir_id int, primary key(mov_id), foreign key(dir_id) references director(dir_id) on delete cascade);
desc movies;

Field	Type	Null	Key	Default	Extra
mov_id	int	NO	PRI	NULL	
mov_title	varchar(30)	YES		NULL	
mov_year	year	YES		NULL	
mov_lang	varchar(10)	YES		NULL	
dir_id	int	YES	MUL	NULL	

create table moviecast(act_id int, mov_id int, part varchar(20), primary key(act_id, mov_id), foreign key(act_id) references actor(act_id) on delete cascade,
foreign key(mov_id) references movies(mov_id) on delete cascade);
desc moviecast;

Field	Type	Null	Key	Default	Extra
act_id	int	NO	PRI	NULL	
mov_id	int	NO	PRI	NULL	
part	varchar(20)	YES		NULL	

create table rating(mov_id int, rev_stars float, primary key(mov_id, rev_stars), foreign key(mov_id) references movies(mov_id) on delete cascade);
desc rating;

Field	Type	Null	Key	Default	Extra
mov_id	int	NO	PRI	NULL	
rev_stars	float	NO	PRI	NULL	

insert into actor values(100, "Leo DiCaprio", 'M');
insert into actor values(101, "Tom Hanks", 'M');
insert into actor values(102, "Tom Cruise", 'M');
insert into actor values(103, "Margot Robbie", 'F');
insert into actor values(104, "Jennifer Aniston", 'F');
insert into actor values(105, "Gal Gadot", 'F');

select * from actor;

act_id	act_name	act_gender
100	Leo DiCaprio	M
101	Tom Hanks	M
102	Tom Cruise	M
103	Margot Robbie	F
104	Jennifer Aniston	F
105	Gal Gadot	F
NULL	NULL	NULL

insert into director values(200, 'Steven Spielberg', '1649503470');
insert into director values(201, 'Alfred Hitchcock', '7989467865');
insert into director values(202, 'James Cameron', '5218281077');
insert into director values(203, 'Kathryn Bigelow', '6157228013');
insert into director values(204, 'Niki Caro', '8976600547');
insert into director values(205, 'Sofia Coppola', '3949875040');
select * from director;

dir_id	dir_name	dir_phone
200	Steven Spielberg	1649503470
201	Alfred Hitchcock	7989467865
202	James Cameron	5218281077
203	Kathryn Bigelow	6157228013
204	Niki Caro	8976600547
205	Sofia Coppola	3949875040
NULL	NULL	NULL

insert into movies values(300, 'Avatar', 2010, 'EN', 202);
insert into movies values(301, 'Dial M For Murder', 1990, 'EN', 201);
insert into movies values(302, 'Jurassic Park 1', 1999, 'EN', 200);
insert into movies values(303, 'Jurassic Park 2', 2017, 'EN', 200);
insert into movies values(304, 'Vertigo', 1986, 'EN', 201);
insert into movies values(305, 'Zero Dark Thirty', 2012, 'EN', 200);
select * from movies;

mov_id	mov_title	mov_year	mov_lang	dir_id
300	Avatar	2010	EN	202
301	Dial M For Murder	1990	EN	201
302	Jurassic Park 1	1999	EN	200
303	Jurassic Park 2	2017	EN	200
304	Vertigo	1986	EN	201
305	Zero Dark Thirty	2012	EN	200
NULL	NULL	NULL	NULL	NULL

```
insert into moviecast values(101, 300, 'actor');
insert into moviecast values(105, 300, 'actress');
insert into moviecast values(102, 301, 'actor');
insert into moviecast values(103, 301, 'actress');
insert into moviecast values(100, 302, 'actor');
insert into moviecast values(104, 302, 'actress');
insert into moviecast values(100, 303, 'actor');
insert into moviecast values(104, 303, 'actress');
insert into moviecast values(102, 304, 'actor');
insert into moviecast values(105, 304, 'actress');
insert into moviecast values(103, 305, 'actress');
select * from moviecast;
```

act_id	mov_id	part
100	302	actor
100	303	actor
101	300	actor
102	301	actor
102	304	actor
103	301	actress
103	305	actress
104	302	actress
104	303	actress
105	300	actress
105	304	actress
NULL	NULL	NULL

```
insert into rating values(300, 4.5);
insert into rating values(301, 3);
insert into rating values(302, 4);
insert into rating values(303, 3.5);
insert into rating values(304, 5);
```

insert into rating values(305, 4);
select * from rating;

mov_id	rev_stars
300	4.5
301	3
302	4
303	3.5
304	5
305	4
NULL	NULL

i. List the titles of all movies directed by 'Hitchcock'.

select m.mov_title from movies m, director d where m.dir_id=d.dir_id and d.dir_name='Alfred Hitchcock';

mov_title
Dial M For Murder
Vertigo

ii. Find the movie names where one or more actors acted in two or more movies.

select m.mov_title from movies m, moviecast mc where m.mov_id=mc.mov_id
and act_id in(select act_id from moviecast group by act_id having count(act_id) > 1) group by m.mov_title having count(*) >= 2;

mov_title
Dial M For Murder
Jurassic Park 1
Jurassic Park 2
Vertigo

iii. List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).

select a.act_name, m.mov_title, m.mov_year from actor a, movies m, moviecast mc
where a.act_id=mc.act_id and mc.mov_id=m.mov_id and m.mov_year not between 2000 and 2015;

act_name	mov_title	mov_year
Tom Cruise	Dial M For Murder	1990
Margot Robbie	Dial M For Murder	1990
Leo DiCaprio	Jurassic Park 1	1999
Jennifer Aniston	Jurassic Park 1	1999
Leo DiCaprio	Jurassic Park 2	2017
Jennifer Aniston	Jurassic Park 2	2017
Tom Cruise	Vertigo	1986
Gal Gadot	Vertigo	1986

iv. Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.

select mov_title, max(rev_stars) from movies inner join rating using (mov_id)
group by mov_title having max(rev_stars)>0 order by mov_title;

mov_title	max(rev_stars)
Avatar	4.5
Dial M For Murder	3
Jurassic Park 1	4
Jurassic Park 2	3.5
Vertigo	5
Zero Dark Thirty	4

v. Update rating of all movies directed by 'Steven Spielberg' to 5.

update rating set rev_stars = 5 where mov_id in (select mov_id from movies where dir_id in
(select dir_id from director where dir_name='Steven Spielberg'));
select * from rating;

mov_id	rev_stars
300	4.5
301	3
302	5
303	5
304	5
305	5
NULL	NULL

PROGRAM 10: College Database

Consider the schema for College Database:

STUDENT (USN, SName, Address, Phone, Gender)

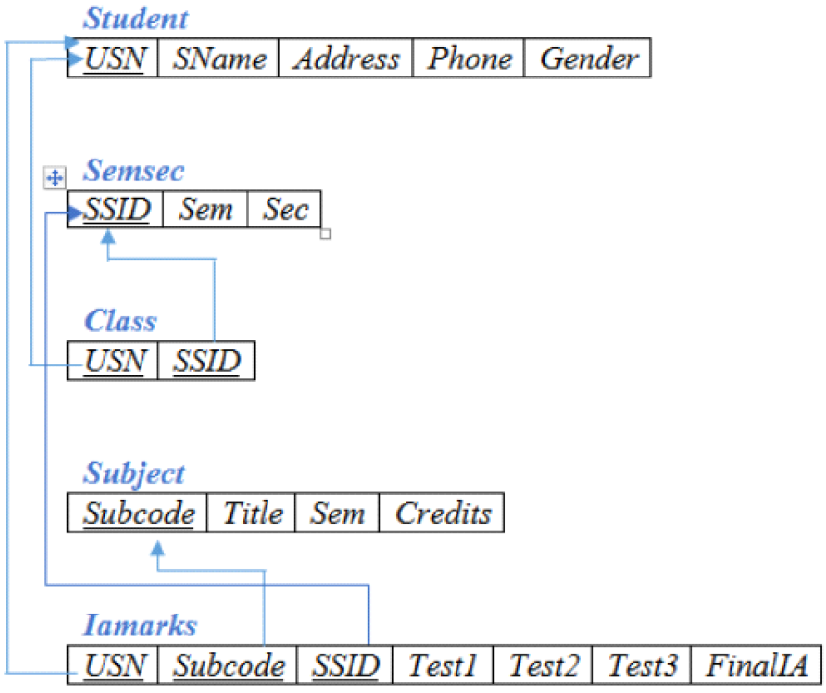
SEMSEC (SSID, Sem, Sec)

CLASS (USN, SSID)

SUBJECT (Subcode, Title, Sem, Credits)

IAMARKS (USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)

Schema Diagram




```
create database Lab10;
use Lab10;
create table student(
  usn varchar(30),
  sname varchar(30),
  address varchar(30),
  phone real,
```

```
 gender varchar(30),
 primary key(usn)
 );
 desc student;
```

	Field	Type	Null	Key	Default	Extra
▶	usn	varchar(30)	NO	PRI	NULL	
	sname	varchar(30)	YES		NULL	
	address	varchar(30)	YES		NULL	
	phone	double	YES		NULL	
	gender	varchar(30)	YES		NULL	

```
create table semsec(
 ssid varchar(30),
 sem int,
 sec varchar(30),
 primary key(ssid)
 );
 desc semsec;
```

	Field	Type	Null	Key	Default	Extra
▶	ssid	varchar(30)	NO	PRI	NULL	
	sem	int	YES		NULL	
	sec	varchar(30)	YES		NULL	

```
 create table class(
        usn varchar(30),
        ssid varchar(30),
        primary key(usn,ssid),
        foreign key(usn) REFERENCES student(usn),
        foreign key(ssid) REFERENCES semsec(ssid)
   );
 desc class;
```

	Field	Type	Null	Key	Default	Extra
▶	usn	varchar(30)	NO	PRI	NULL	
	ssid	varchar(30)	NO	PRI	NULL	

```
create table subject(
  code varchar(30),
  title varchar(30),
  sem int,
  credits int,
  primary key(code)
```

```
  );
desc subject;
```

	Field	Type	Null	Key	Default	Extra
▶	code	varchar(30)	NO	PRI	NULL	
	title	varchar(30)	YES		NULL	
	sem	int	YES		NULL	
	credits	int	YES		NULL	

```
create table marks(
        usn varchar(30),code varchar(30),
        ssid varchar(30),
        test1 real, test2 real, test3 real, final real,
        primary key(usn,code,ssid),
        foreign key(usn) REFERENCES student(usn),
        foreign key(code) REFERENCES subject(code),
        foreign key(ssid) REFERENCES semsec(ssid)
   );
desc marks;
```

	Field	Type	Null	Key	Default	Extra
▶	usn	varchar(30)	NO	PRI	NULL	
	code	varchar(30)	NO	PRI	NULL	
	ssid	varchar(30)	NO	PRI	NULL	
	test1	double	YES		NULL	
	test2	double	YES		NULL	
	test3	double	YES		NULL	
	final	double	YES		NULL	

```
insert into student values('1RN13CS020','akshay','belagavi',8877881122,'m'),
('1RN13CS062','sandhya','bengaluru',7722829912,'f'),
('1RN13CS091','teesha','bengaluru',7712312312,'f'),
('1RN13CS066','supriya','mangaluru',8877881122,'f'),
('1RN14CS010','abhay','bengaluru',9900211201,'m'),
('1RN14CS032','bhaskar','bengaluru',9923211099,'m'),
('1RN14CS025','asmi','bengaluru',7894737377,'f'),
('1RN15CS011','ajay','tumkur',98545091341,'m'),
('1RN15CS029','chitra','davangere',7696772121,'f'),
('1RN15CS045','jeeva','bellary',9944850121,'m'),
('1RN15CS091','santosh','mangaluru',8812332201,'m'),
('1RN16CS045','ismail','kalburgi',9900232201,'m'),
('1RN16CS088','sameera','shimoga',9905542212,'f'),
('1RN16CS122','vinayaka','chikamagaluru',8800880011,'m');
select * from student;
```

usn	sname	address	phone	gender
1RN13CS020	akshay	belagavi	8877881122	m
1RN13CS062	sandhya	bengaluru	7722829912	f
1RN13CS066	supriya	mangaluru	8877881122	f
1RN13CS091	teesha	bengaluru	7712312312	f
1RN14CS010	abhay	bengaluru	9900211201	m
1RN14CS025	asmi	bengaluru	7894737377	f
1RN14CS032	bhaskar	bengaluru	9923211099	m
1RN15CS011	ajay	tumkur	98545091341	m
1RN15CS029	chitra	davangere	7696772121	f
1RN15CS045	jeeva	bellary	9944850121	m
1RN15CS091	santosh	mangaluru	8812332201	m
1RN16CS045	ismail	kalburgi	9900232201	m
1RN16CS088	sameera	shimoga	9905542212	f
1RN16CS122	vinayaka	chikamag...	8800880011	m
NULL	NULL	NULL	NULL	NULL

insert into semsec values('CSE8A',8,'A'),
('CSE8B',8,'B'),('CSE8C',8,'C'),
('CSE7A',7,'A'),('CSE7B',7,'B'),('CSE7C',7,'C'),
('CSE6A',6,'A'),('CSE6B',6,'B'),('CSE6C',6,'C'),
('CSE5A',5,'A'),('CSE5B',5,'B'),('CSE5C',5,'C'),
('CSE4A',4,'A'),('CSE4B',4,'B'),('CSE4C',4,'C'),
('CSE3A',3,'A'),('CSE3B',3,'B'),('CSE3C',3,'C'),
('CSE2A',2,'A'),('CSE2B',2,'B'),('CSE2C',2,'C'),
('CSE1A',1,'A'),('CSE1B',1,'B'),('CSE1C',1,'C');
select * from semsec;

ssid	sem	sec
CSE1A	1	A
CSE1B	1	B
CSE1C	1	C
CSE2A	2	A
CSE2B	2	B
CSE2C	2	C
CSE3A	3	A
CSE3B	3	B
CSE3C	3	C
CSE4A	4	A
CSE4B	4	B
CSE4C	4	C
CSE5A	5	A
CSE5B	5	B
CSE5C	5	C

	CSE6A	6	A
	CSE6B	6	B
	CSE6C	6	C
	CSE7A	7	A
	CSE7B	7	B
	CSE7C	7	C
	CSE8A	8	A
	CSE8B	8	B
	CSE8C	8	C
*	NULL	NULL	NULL

insert into class values('1RN13CS020','CSE8A'),
('1RN13CS062','CSE8A'),('1RN13CS066','CSE8B'),('1RN13CS091','CSE8C'),
('1RN14CS010','CSE7A'),('1RN14CS025','CSE7A'),('1RN14CS032','CSE7A'),
('1RN15CS011','CSE4A'),('1RN15CS029','CSE4A'),('1RN15CS045','CSE4B'),
('1RN15CS091','CSE4C'),('1RN16CS045','CSE3A'),('1RN16CS088','CSE3B'),
('1RN16CS122','CSE3C');
select * from class;

	usn	ssid
▶	1RN16CS045	CSE3A
	1RN16CS088	CSE3B
	1RN16CS122	CSE3C
	1RN15CS011	CSE4A
	1RN15CS029	CSE4A
	1RN15CS045	CSE4B
	1RN15CS091	CSE4C
	1RN14CS010	CSE7A
	1RN14CS025	CSE7A
	1RN14CS032	CSE7A
	1RN13CS020	CSE8A
	1RN13CS062	CSE8A
	1RN13CS066	CSE8B
	1RN13CS091	CSE8C
*	NULL	NULL

insert into subject values('10CS81','ACA',8,4),
('10CS82','SSM',8,4),('10CS83','NM',8,4),
('10CS84','CC',8,4),('10CS85','PW',8,4),
('10CS71','OOAD',7,4),('10CS72','ECS',7,4),
('10CS73','PTW',7,4),('10CS74','DWDM',7,4),
('10CS75','JAVA',7,4),('10CS76','SAN',7,4),
('10CS51','ME',5,4),('10CS52','CN',5,4),

('10CS53','DBMS',5,4),('10CS54','ATC',5,4),
('10CS55','JAVA',5,3),('10CS56','AI',5,3),
('10CS41','M4',4,4),('10CS42','SE',4,4),
('10CS43','DAA',4,4),('10CS44','MPMC',4,4),
('10CS45','OOC',4,3),('10CS46','DC',4,3),
('10CS31','M3',3,4),('10CS32','ADE',3,4),
('10CS33','DSA',3,4),('10CS34','CO',3,4),
('10CS35','USP',3,3),('10CS36','DMS',3,3);
select * from subject;

code	title	sem	credits
10CS31	M3	3	4
10CS32	ADE	3	4
10CS33	DSA	3	4
10CS34	CO	3	4
10CS35	USP	3	3
10CS36	DMS	3	3
10CS41	M4	4	4
10CS42	SE	4	4
10CS43	DAA	4	4
10CS44	MPMC	4	4
10CS45	OOC	4	3
10CS46	DC	4	3
10CS51	ME	5	4
10CS52	CN	5	4
10CS53	DBMS	5	4
10CS54	ATC	5	4
10CS55	JAVA	5	3
10CS56	AI	5	3
10CS71	OOAD	7	4
10CS72	ECS	7	4
10CS73	PTW	7	4
10CS74	DWDM	7	4
10CS75	JAVA	7	4
10CS76	SAN	7	4
10CS81	ACA	8	4
10CS82	SSM	8	4
10CS83	NM	8	4
10CS84	CC	8	4
10CS85	PW	8	4
NULL	NULL	NULL	NULL

insert into marks(usn,code,ssid,test1,test2,test3)
values('1RN13CS091','10CS81','CSE8C',15,16,18),
('1RN13CS091','10CS82','CSE8C',12,19,14),('1RN13CS091','10CS83','CSE8C',19,15,
20),

('1RN13CS091','10CS84','CSE8C',20,16,19),('1RN13CS091','10CS85','CSE8C',15,15,12);

select * from marks;

usn	code	ssid	test1	test2	test3	final
1RN13CS091	10CS81	CSE8C	15	16	18	NULL
1RN13CS091	10CS82	CSE8C	12	19	14	NULL
1RN13CS091	10CS83	CSE8C	19	15	20	NULL
1RN13CS091	10CS84	CSE8C	20	16	19	NULL
1RN13CS091	10CS85	CSE8C	15	15	12	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

i. List all the student details studying in fourth semester 'C' section.

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';

usn	sname	address	phone	gender	sem	sec
1RN15CS091	santosh	mangaluru	8812332201	m	4	C

ii. Compute the total number of male and female students in each semester and in each section.

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;

sem	sec	gender	COUNT
3	A	m	1
3	B	f	1
3	C	m	1
4	A	f	1
4	A	m	1
4	B	m	1
4	C	m	1
7	A	f	1
7	A	m	2
8	A	f	1
8	A	m	1
8	B	f	1
8	C	f	1

iii. Create a view of Test1 marks of student USN '1BI15CS101' in all subjects.

```
create view STU_test1_marks_view as
select test1, code
from marks
where usn = '1RN13CS091';
select * from STU_test1_marks_view;
```

test1	code
15	10CS81
12	10CS82
19	10CS83
20	10CS84
15	10CS85

iv. Categorize students based on the following criterion:

If FinalIA = 17 to 20 then CAT = 'Outstanding'
If FinalIA = 12 to 16 then CAT = 'Average'
If FinalIA< 12 then CAT = 'Weak'
Give these details only for 8th semester A, B, and C section students.

```
select S.usn, S.sname, S.address, S.phone, S.gender,
(CASE
when IA.final between 17 and 20 then 'outstanding'
when IA.final between 12 and 16 then 'average'
else 'weak' end) AS CAT
from student S, semsec SS, marks IA, subject sub
where S.usn = IA.usn AND SS.ssid = IA.ssid AND sub.code = IA.code AND sub.sem
= 8;
```

usn	sname	address	phone	gender	CAT
1RN13CS091	teesha	bengaluru	7712312312	f	weak
1RN13CS091	teesha	bengaluru	7712312312	f	weak
1RN13CS091	teesha	bengaluru	7712312312	f	weak
1RN13CS091	teesha	bengaluru	7712312312	f	weak
1RN13CS091	teesha	bengaluru	7712312312	f	weak