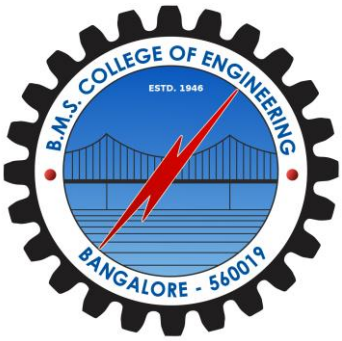
B.M.S. COLLEGE OF ENGINEERING

(AUTONOMOUS COLLEGE UNDER VTU)
BENGALURU-19



LAB TEST REPORT

NAME : Mazin Salim
USN : 1BM19CS201
COURSE NAME : DATABASE MANAGEMENT SYSTEMS
COURSE TITLE : 19CS4PCDBM
SEM : 4
SECTION : D

LAB PROGRAMS 1-10:

PROGRAM 1: INSURANCE DATABASE

Consider the Insurance database given below. The primary keys are underlined and the data types are specified.
PERSON (driver-id #: String, name: String, address: String)

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

ACCIDENT (report-number: int, date: date, location: String)

OWNS (driver-id #: String, Regno: String)

PARTICIPATED (driver-id: String, Regno: String, report-number: int, damage-amount: int)

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

```
create database INSURANCE;
create table INSURANCE.person(
driver_id varchar(10),
name varchar(20),
address varchar(30),
primary key(driver_id)
);
create table INSURANCE.car(
reg_num varchar(10),
model varchar(10),
year int,
primary key(reg_num)
);
create table INSURANCE.accident(
report_num int,
accident_date date,
location varchar(20),
primary key(report_num)
);
create table INSURANCE.owns(
driver_id varchar(10),
reg_num varchar(10),
primary key(driver_id,reg_num),
foreign key(driver_id) references person(driver_id),
foreign key(reg_num) references car(reg_num)
);
create table INSURANCE.participated(
driver_id varchar(10),
reg_num varchar(10),
report_num int,
damage_amount int,
primary key(driver_id,reg_num,report_num),
foreign key(driver_id) references person(driver_id),
foreign key(reg_num) references car(reg_num),
foreign key(report_num) references accident(report_num));
```

ii. Enter at least five tuples for each relation

```
use INSURANCE;
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');
select * from person;
```

driver_id	name	address
A01	Richard	Srinivas Nagar
A02	Pradeep	Rajajinagar
A03	Smith	Ash
A04	Venu	N.R.Colony
A05	John	Hanumanth Nagar
NULL	NULL	NULL

Rajajinagar

```
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);
select * from car;
```

reg_num	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,'2001-01-03','Mysore Road');
insert into accident values(12,'2002-01-04','Southend Circle');
insert into accident values(13,'2021-01-03','Bulltemple Road');
insert into accident values(14,'2017-02-08','Mysore Road');
insert into accident values(15,'2008-03-05 ','Kanakpura Road');
select * from accident;
```

report_num	accident_date	location
11	2001-01-03	Mysore Road
12	2002-01-04	Southend Circle
13	2021-01-03	Bulltemple Road
14	2017-02-08	Mysore Road
15	2008-03-05	Kanakpura Road
NULL	NULL	NULL

```
insert into owns values('A01','KA052250');
insert into owns values('A02','KA053408');
insert into owns values('A03','KA031181');
insert into owns values('A04','KA095477');
insert into owns values('A05','KA041702');
select * from owns;
```

driver_id	reg_num
A03	KA031181
A05	KA041702
A01	KA052250
A02	KA053408
A04	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);
select * from participated;
```

driver_id	reg_num	report_num	damage_amount
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 for the car with a specific Regno in the accident with report number 12 to 25000.
b. Add a new accident to the database.

```
use INSURANCE;
UPDATE participated
SET damage_amount=25000
WHERE report_num=12;
insert into accident values('16','2009-04-05','Mysore Road');
select * from accident;
```

report_num	accident_date	location
11	2001-01-03	Mysore Road
12	2002-01-04	Southend Circle
13	2021-01-03	Bulltemple Road
14	2017-02-08	Mysore Road
15	2008-03-05	Kanakpura Road
16	2009-04-05	Mysore Road
NULL	NULL	NULL

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

```
use INSURANCE;
SELECT COUNT(DISTINCT driver_id) FROM accident, participated
WHERE accident.report_num = participated.report_num
AND accident_date LIKE '2008%';
```

	COUNT(DISTINCT driver_id)
▶	1

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

```
use INSURANCE;
SELECT COUNT(report_num) FROM car, participated
WHERE car.reg_num = participated.reg_num
AND model='Lancer';
```

	COUNT(report_num)
▶	1

PROGRAM 2: BANKING ENTERPRISE DATABASE

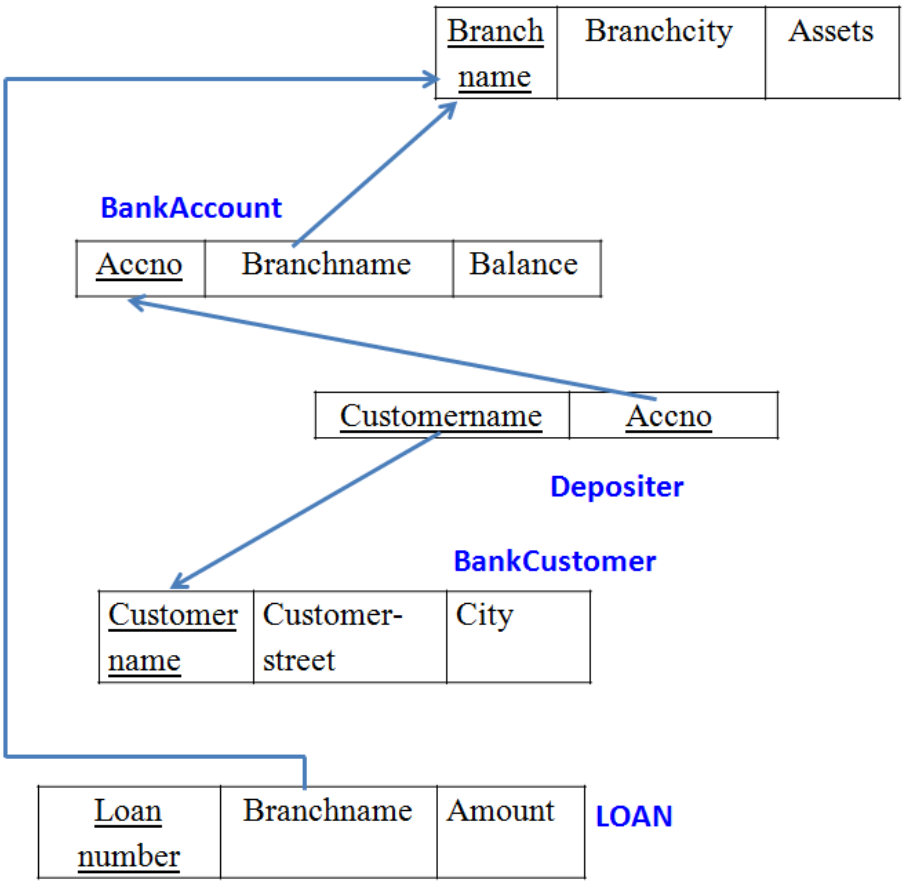
Consider the following database for a banking enterprise.

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)

i. Create the above tables by properly specifying the primary keys and the foreign keys.
ii. Enter at least five tuples for each relation.
iii. Find all the customers who have at least two accounts at the *Main* branch (ex. SBI_ResidencyRoad).
iv. Find all the customers who have an account at *all* the branches located in a specific city (Ex. Delhi).
v. Demonstrate how you delete all account tuples at every branch located in a specific city (Ex. Bombay).

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(
branch_name varchar(30),
branch_city varchar(30),
assests real,
primary key(branch_name));
create table BankCustomer(
customer_name varchar(30),
customer_street varchar(30),
customer_city varchar(30),
primary key(customer_name));
create table BankAccount(
accno int,
branch_name varchar(20),
balance real,
primary key(accno),
foreign key(branch_name) references Branch(branch_name));
create table Depositer(
customer_name varchar(20),
accno int,
primary key(customer_name,accno),
foreign key(customer_name) references BankCustomer(customer_name),
foreign key(accno) references BankAccount(accno)
);
create table Loan(
loan_number int,
branch_name varchar(20),
Amount real,
primary key(loan_number),
foreign key(branch_name) references Branch(branch_name)
);
```

ii. Enter at least five tuples for each relation.

```
use Bank;
insert into Branch values('SBI_Chamrajpet','Bangalore',50000);
insert into Branch values('SBI_ResidencyRoad','Bangalore',10000);
insert into Branch values('SBI_ShivajiRoad','Bangalore',20000);
```

```
insert into Branch values('SBI_ParliamentRoad','Delhi',10000);
insert into Branch values('SBI_Jantarmantar','Delhi',20000);
select *from Branch;
```

branch_name	branch_city	assests
SBI_Chamrajpet	Bangalore	50000
SBI_Jantarmantar	Delhi	20000
SBI_ParlimentRoad	Delhi	10000
SBI_ResidencyRoad	Bangalore	10000
SBI_ShivajiRoad	Bangalore	20000
NULL	NULL	NULL

```
insert into Loan values(2,'SBI_ResidencyRoad',2000);
insert into Loan values(1,'SBI_Chamrajpet',1000);
insert into Loan values(3,'SBI_ShivajiRoad',3000);
insert into Loan values(4,'SBI_ParliamentRoad',4000);
insert into Loan values(5,'SBI_Jantarmantar',3000);
select *from Loan;
```

loan_number	branch_name	Amount
1	SBI_Chamrajpet	1000
2	SBI_ResidencyRoad	2000
3	SBI_ShivajiRoad	3000
4	SBI_ParlimentRoad	4000
5	SBI_Jantarmantar	3000
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_ParliamentRoad',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_ParliamentRoad', 3000);
insert into BankAccount values(10, 'SBI_ResidencyRoad', 5000);
insert into BankAccount values(11, 'SBI_Jantarmantar', 2000);
select *from BankAccount;
```

accno	branch_name	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', 'National_College_Road', 'Bangalore');
insert into BankCustomer values ('Nikhil', 'Akbar_Road', 'Delhi');
insert into BankCustomer values ('Ravi', 'Prithviraj_Road', 'Delhi');
select *from BankCustomer;
```

customer_name	customer_street	customer_city
Avinash	Bull_Temple_Road	Bangalore
Dinesh	Bannergatta_Road	Bangalore
Mohan	National_College_Road	Bangalore
Nikhil	Akbar_Road	Delhi
Ravi	Prithviraj_Road	Delhi
NULL	NULL	NULL

```
insert into Depositer values('Avinash', 1);
insert into Depositer values('Dinesh', 2);
insert into Depositer values('Nikhil', 4);
insert into Depositer values('Ravi', 5);
insert into Depositer values('Avinash', 8);
insert into Depositer values('Nikhil', 9);
insert into Depositer values('Dinesh', 10);
insert into Depositer values('Nikhil', 11);
select *from Depositer;
```

customer_name	accno
Avinash	1
Dinesh	2
Nikhil	4
Ravi	5
Avinash	8
Nikhil	9
Dinesh	10
Nikhil	11
NULL	NULL

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

```
use Bank;
select c.customer_name
from BankCustomer c
where exists(
select d.customer_name
from Depositer d, BankAccount ba
where d.accno=ba.accno
and c.customer_name=d.customer_name
and ba.branch_name='SBI_ResidencyRoad'
group by d.customer_name
having count(d.customer_name)>=2);
```

	customer_name
▶	Dinesh
*	NULL

iv. Find all the customers who have an account at *all* the branches located in a specific city (Ex. Delhi).

```
use Bank;
select distinct d.customer_name
from Depositer d
where exists(
select * from BankAccount ba
where ba.accno=d.accno
and exists(
select * from Branch b
where b.branch_name = ba.branch_name
and b.branch_city='Delhi'));
```

	customer_name
▶	Ravi
	Nikhil

v. Demonstrate how you delete all account tuples at every branch located in a specific city (Ex. Bombay).

```
use Bank;
delete from BankAccount
where branch_name in(
select branch_name
from branch
where branch_city = 'Bombay');
select *from BankAccount;
```

	accno	branch_name	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

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.

Schema Diagram

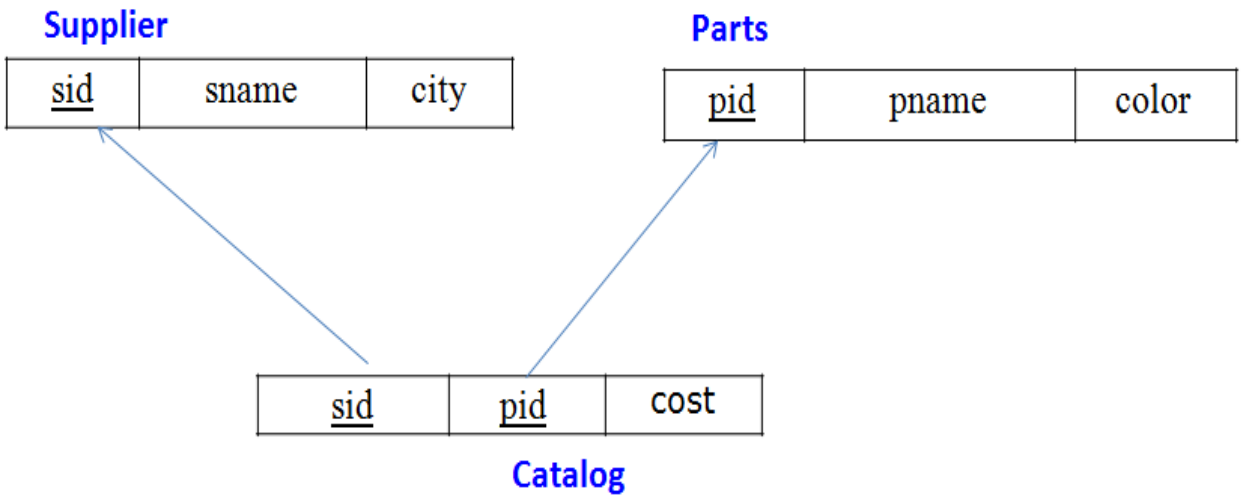


Table Data

SUPPLIERS

SID	SNAME	CITY
10001	Acme Widget	Bangalore
10002	Johns	Kolkata
10003	Vimal	Mumbai
10004	Reliance	Delhi

PARTS

PID	PNAME	COLOR
20001	Book	Red
20002	Pen	Red
20003	Pencil	Green
20004	Mobile	Green
20005	Charger	Black

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

QUERY 1

```
create database Suppliers;
use Suppliers;
create table supplier(
sid int,
sname char(30),
city char(30));
create table catalog(
sid int,
pid int,
cost int);
create table parts(
pid int,
pname varchar(15),
color varchar(10),
primary key (pid));
```

QUERY 2

```
use Suppliers;
insert into supplier values(10001, 'Acme Widget','Bengaluru');
insert into supplier values(10002,'Johns','Kolkata');
insert into supplier values(10003, 'Vimal','Mumbai');
insert into supplier values(10004, 'Reliance','Delhi');
insert into supplier values(10005, 'Mahindra','Mumbai');
select * from supplier;
```

	sid	sname	city
▶	10001	Acme Widget	Bengaluru
	10002	Johns	Kolkata
	10003	Vimal	Mumbai
	10004	Reliance	Delhi
	10005	Mahindra	Mumbai
	10001	Acme Widget	Bengaluru
	10002	Johns	Kolkata
	10003	Vimal	Mumbai
	10004	Reliance	Delhi
	10005	Mahindra	Mumbai

```
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');
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');
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

1.Find the pnames of parts for which there is some supplier.

```
use Suppliers;
select distinct p.pname
from parts p,catalog c
where p.pid = c.pid;
```

	pname
▶	Book
	Pen
	Pencil
	Mobile
	Charger

2.Find the snames of suppliers who supply every part.

```
use Suppliers;
```

```
select s.sname from supplier s
where not exists(
select p.pid from parts p
where not exists(
select c.sid from catalog c
where c.sid = s.sid
and c.pid = p.pid));
```

	sname
▶	Acme Widget
	Acme Widget

3.Find the snames of suppliers who supply every red part.

```
use Suppliers;
select s.sname from supplier s
where not exists(
select p.pid from parts p
where p.color = 'Red'
and not exists (
select c.sid from catalog c
where c.sid = s.sid
and c.pid = p.pid));
```

	sname
▶	Acme Widget
	Johns
	Acme Widget
	Johns

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

```
use Suppliers;
select p.pname from parts p,catalog c,supplier s
where p.pid = c.pid and c.sid = s.sid and s.sname = 'Acme Widget'
and not exists(
select * from catalog c1, supplier s1
where p.pid = c1.pid
and c1.sid = s1.sid
and s1.sname <> 'Acme Widget');
```

	pname
▶	Mobile
	Mobile
	Charger
	Charger

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).

```
use Suppliers;
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

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

```
use Suppliers;
select p.pid, s.sname
from parts p, supplier 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
	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, meetsat: time, room: string, fid: integer)

ENROLLED(snum: integer, cname:string)

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

```
create database StudentFaculty;
use StudentFaculty;
create table student(
snum int,
sname varchar(10),
major varchar(2),
lvl varchar(2),
age int,
primary key (snum));
create table faculty(
fid int,
fname varchar(20),
deptid int,
primary key(fid));
create table class(
cname varchar(20),
meetsat timestamp,
room varchar(10),
fid int,
primary key (cname),
foreign key(fid) references faculty(fid));
create table enrolled(
snum int,
cname varchar(20),
primary key(snum,cname),
foreign key(snum) references student(snum),
foreign key(cname) references class(cname));
```

```
use StudentFaculty;
insert into student values(1, 'jhon', '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);
select * from student;
```

snum	sname	major	lvl	age
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
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);
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', 12);
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);
select * from class;
```

cname	meetsat	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	12
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');
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 Harish

```
use StudentFaculty;
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.meetsat = C2.meetsat);
```

	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.

```
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: AIRLNE 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.

```
create database flightdb;
use flightdb;
create table flights(
flno int,
fromplace varchar(15),
toplace varchar(15),
distance int,
departs datetime,
arrives datetime,
price int,
primary key (flno));
create table aircraft(
aid int,
aname varchar(15),
cruisingrange int,
primary key (aid));
create table employees (
eid int,
ename varchar(15),
salary int,
primary key (eid));
create table certified (
eid int,
aid int,
foreign key (eid) references employees(eid),
foreign key (aid) references aircraft(aid));
```

```
use flightdb;
insert into flights values(101, 'Bangalore', 'Delhi', 2500, '2005-05-13 07:15:31', '2005-05-13 18:15:31',
5000);
```

```
insert into flights values(102, 'Bangalore', 'Lucknow', 3000, '2013-05-05 07:15:31', '2013-05-05 11:15:31',
6000);
insert into flights values(103, 'Lucknow', 'Delhi', 500, '2013-05-05 12:15:31', '2013-05-05 17:15:31',
3000);
insert into flights values(107, 'Bangalore', 'Frankfurt', 8000, '2013-05-05 07:15:31', '2013-05-05 22:15:31',
60000);
insert into flights values(104, 'Bangalore', 'Frankfurt', 8500, '2013-05-05 07:15:31', '2013-05-05 23:15:31',
75000);
insert into flights values(105, 'Kolkata', 'Delhi', 3400, '2013-05-05 07:15:31', '2013-05-05 09:15:31',
7000);
insert into flights values(106, 'Bangalore', 'Kolkata', 1000, '2013-05-05 01:15:30', '2013-05-05 09:20:30',
10000);
insert into flights values(108, 'Lucknow', 'Kolkata', 1000, '2013-05-05 11:30:30', '2013-05-05 15:20:30',
10000);
select * from flights;
```

flno	fromplace	toplace	distance	departs	arrives	price
101	Bangalore	Delhi	2500	2005-05-13 07:15:31	2005-05-13 18:15:31	5000
102	Bangalore	Lucknow	3000	2013-05-05 07:15:31	2013-05-05 11:15:31	6000
103	Lucknow	Delhi	500	2013-05-05 12:15:31	2013-05-05 17:15:31	3000
104	Bangalore	Frankfurt	8500	2013-05-05 07:15:31	2013-05-05 23:15:31	75000
105	Kolkata	Delhi	3400	2013-05-05 07:15:31	2013-05-05 09:15:31	7000
106	Bangalore	Kolkata	1000	2013-05-05 01:15:30	2013-05-05 09:20:30	10000
107	Bangalore	Frankfurt	8000	2013-05-05 07:15:31	2013-05-05 22:15:31	60000
108	Lucknow	Kolkata	1000	2013-05-05 11:30:30	2013-05-05 15:20:30	10000
NULL	NULL	NULL	NULL	NULL	NULL	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);
insert into aircraft values(108, '707', 760);
insert into aircraft values(109, '747', 1000);
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
108	707	760
109	747	1000
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(702, 108);
insert into certified values(701, 109);
select * from certified;
```

eid	aid
701	101
701	102
701	106
701	105
702	104
703	104
704	104
702	107
703	107
704	107
702	101
702	108
701	109

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

```
use flightdb;
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
	Dream
	707

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.

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

	max(a.cruisingrange)	eid
▶	3500	701
	120000	702

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

use flightdb;
select ename from employees
where salary <(select min(price)
from flights
where fromplace='Bangalore' and toplace='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.

use flightdb;
select avg(e.salary), c.aid from certified c, employees e
where c.aid in(select aid
from aircraft where cruisingrange>1000)
and e.eid = c.eid group by c.aid;

	avg(e.salary)	aid
▶	75000.0000	101
	113333.3333	104
	50000.0000	105
	50000.0000	106
	113333.3333	107

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

```
use flightdb;
select ename from employees where eid in(
select eid from certified where aid in(
select aid from aircraft where aname = 'Boeing'));
```

	ename
▶	A

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

```
use flightdb;
select aname from aircraft
where cruisingrange > any(select distance
from flights where fromplace='Bangalore' and toplace='Delhi');
```

	aname
▶	747
	Dreamliner
	Boeing
	Dream

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

```
use flightdb;
select F.flno, F.departs from flights F
where F.flno in ((select F0.flno
from flights F0
where F0.fromplace = 'Bangalore' and F0.toplace = 'Kolkata'
and EXTRACT(hour from F0.arrives) < 18 )
UNION(select F0.flno
from flights F0, flights F1
where F0.fromplace = 'Bangalore' and F0.toplace <> 'Kolkata'
and F0.toplace = F1.fromplace and F1.toplace = 'Kolkata'
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.fromplace = 'Bangalore'
and F0.toplace = F1.fromplace
and F1.toplace = F2.fromplace
and F2.toplace = 'Kolkata'
and F0.toplace <> 'Kolkata'
and F1.toplace <> 'Kolkata'
and F1.departs > F0.arrives
and F2.departs > F1.arrives
and EXTRACT(hour from F2.arrives) < 18))
```

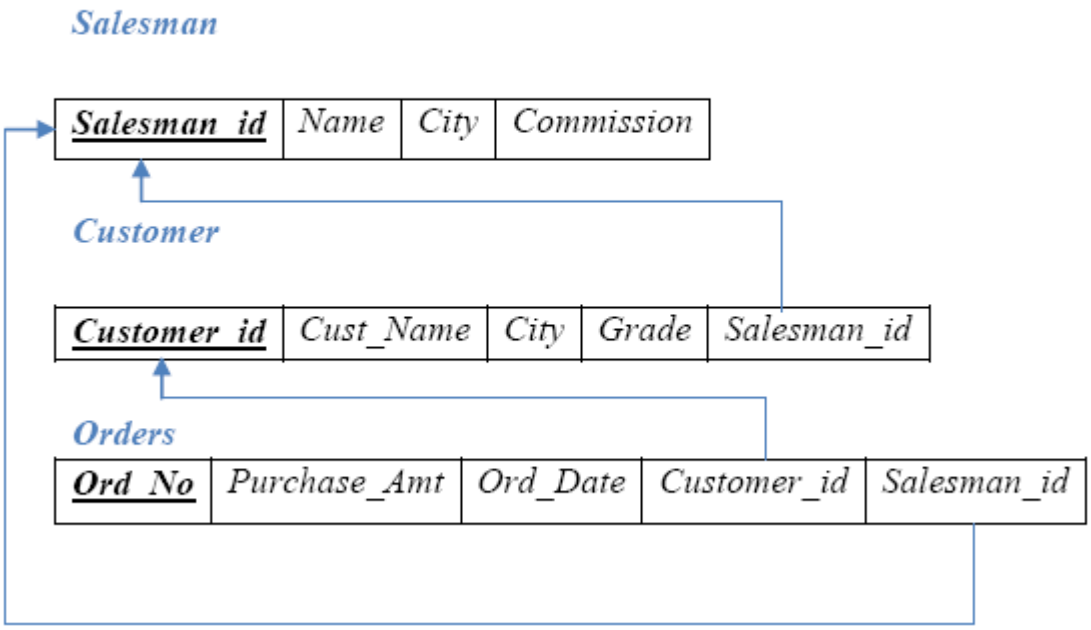
	flno	departs
▶	102	2013-05-05 07:15:31
	106	2013-05-05 01:15:30

PROGRAM 6: ORDER DATABASE

Q.Consider the following schema for 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



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

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

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

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

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

QUERY 1

```
create database order_database;
use order_database;

create table salesman(
Salesman_id int,
Name char(30),
City varchar(30),
Commission varchar(5),
primary key(Salesman_id));

create table customer(
Customer_id int,
Cust_Name char(30),
City varchar(30),
Grade int,
Salesman_id int,
primary key(Customer_id,Salesman_id),
foreign key(Salesman_id) references salesman(Salesman_id) on delete cascade);

create table orders(
Ord_No int,
Purchase_Amt int,
Ord_Date varchar(20),
Customer_id int,
Salesman_id int,
primary key(Customer_id,Salesman_id),
foreign key(Customer_id) references customer(Customer_id) on delete cascade,
foreign key(Salesman_id) references salesman(Salesman_id) on delete cascade);
```

QUERY 2

use order_database;

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','Hyderaba','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	Hyderaba	15%
NULL	NULL	NULL	NULL

insert into customer values('10','Preethi','Bangalore','100','1000');

insert into customer values('11','Vivek','Bangalore','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	Bangalore	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-MAY-17','10','1000');

insert into orders values('51','450','20-JAN-17','10','2000');

insert into orders values('52','1000','24-FEB-17','13','2000');

insert into orders values('53','3500','13-APR-17','14','3000');

insert into orders values('54','550','09-MAR-17','12','2000');

select * from orders;

Ord_No	Purchase_Amt	Ord_Date	Customer_id	Salesman_id
50	5000	04-MAY-17	10	1000
51	450	20-JAN-17	10	2000
54	550	09-MAR-17	12	2000
52	1000	24-FEB-17	13	2000
53	3500	13-APR-17	14	3000
NULL	NULL	NULL	NULL	NULL

QUERY 3

use order_database;

select grade,COUNT(*)

from customer

group by grade

having grade > (select avg(grade)

from customer

where city = 'Bangalore');

grade	COUNT(*)
300	1
400	2

QUERY 4

use order_database;

select Name,Salesman_id from salesman s

where 1<(select count(*)

from customer

where Salesman_id=s.Salesman_id);

	Name	Salesman_id
▶	John	1000
	Ravi	2000
*	NULL	NULL

QUERY 5

use order_database;

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);

	salesman_id	name	cust_name	commission
▶	1000	John	Preethi	25%
	2000	Ravi	Preethi	20%
	1000	John	Vivek	25%
	2000	Ravi	Vivek	20%
	1000	John	Chethan	25%
	2000	Ravi	Chethan	20%
	1000	John	Mamatha	25%
	2000	Ravi	Mamatha	20%
	3000	Kumar	no customer	15%
	4000	Smith	no customer	30%
	5000	Harsha	no customer	15%

QUERY 6

```
use order_database;

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
04-MAY-17	1000	John
20-JAN-17	2000	Ravi
09-MAR-17	2000	Ravi
24-FEB-17	2000	Ravi
13-APR-17	3000	Kumar

QUERY 7

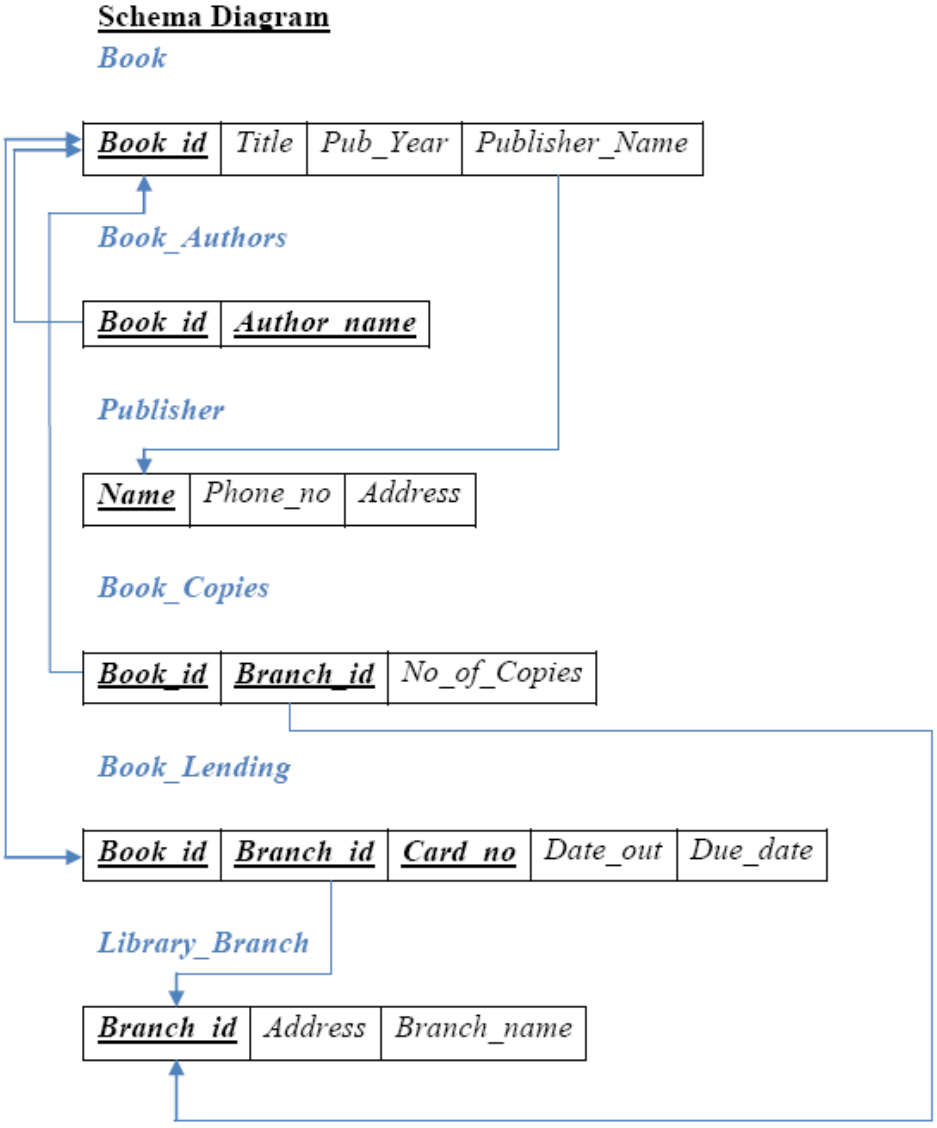
```
use order_database;

delete from salesman
where Salesman_id = 1000;
select * from salesman;
select * from orders;
```

Ord_No	Purchase_Amt	Ord_Date	Customer_id	Salesman_id
54	550	09-MAR-17	12	2000
52	1000	24-FEB-17	13	2000
53	3500	13-APR-17	14	3000
NULL	NULL	NULL	NULL	NULL

PROGRAM 7: BOOK 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



1. Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each branch, etc.
2. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017
3. Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation.
4. Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.

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

QUERY 1

```
create database book_database;
use book_database;

create table publisher (
name varchar(30),
phone int,
address varchar(50),
primary key(name));

create table book (
book_id int,
title varchar(50),
pub_year varchar(10),
publisher_name varchar(30),
primary key(book_id),
foreign key (publisher_name) references publisher (name) on delete cascade
);

create table book_authors (
author_name varchar(30),
book_id int,
foreign key (book_id) references book (book_id) on delete cascade,
primary key (book_id, author_name));
```

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

create table book_copies (
no_of_copies int,
book_id int,
branch_id int,
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));

create table card (
card_no int,
primary key(card_no));

create table book_lending (
date_out date,
due_date date,
book_id int,
branch_id int,
card_no int,
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));
```

QUERY 2

use book_database;

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, 'chenai');

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;
```

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

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

QUERY 3

```
use book_database;

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;
```

	grade	COUNT(*)
▶	300	1
	400	2

QUERY 4

```
use book_database;

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;
```

	Name	Salesman_id
▶	John	1000
	Ravi	2000
*	NULL	NULL

QUERY 5

use book_database;

delete from book where book_id = 3;

select * from book;

select * from book_authors;

select * from book_copies;

select * from book_lending;

salesman_id	name	cust_name	commission
1000	John	Preethi	25%
2000	Ravi	Preethi	20%
1000	John	Vivek	25%
2000	Ravi	Vivek	20%
1000	John	Chethan	25%
2000	Ravi	Chethan	20%
1000	John	Mamatha	25%
2000	Ravi	Mamatha	20%
3000	Kumar	no customer	15%
4000	Smith	no customer	30%
5000	Harsha	no customer	15%

QUERY 6

use book_database;

create view view1

as select pub_year from book;

select * from view1;

ord_date	salesman_id	name
04-MAY-17	1000	John
20-JAN-17	2000	Ravi
09-MAR-17	2000	Ravi
24-FEB-17	2000	Ravi
13-APR-17	3000	Kumar

QUERY 7

```
use book_database;


create view view2
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 view2;
```

Ord_No	Purchase_Amt	Ord_Date	Customer_id	Salesman_id
54	550	09-MAR-17	12	2000
52	1000	24-FEB-17	13	2000
53	3500	13-APR-17	14	3000
NULL	NULL	NULL	NULL	NULL

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)

Database applications laboratory GCEM DEPARTMENT OF CSE Page - 5 - 5th semester
i. Create the above tables by properly specifying the primary keys and the foreign keys.
ii. Enter at least five tuples for each relation.
iii. Demonstrate how you add a new text book to the database and make this book be adopted by some department.
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.
v. List any department that has all its adopted books published by a specific publisher.
vi. Generate suitable reports.
vii. Create suitable front end for querying and displaying the results.

QUERY 1

```
create database StudentEnrollment;
use STudentEnrollment;

create table student(
regno varchar(20),
name varchar(30),
major varchar(20),
bdate date,
primary key (regno)
);

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

create table enroll(
regno varchar(20),
courseno int,
sem int,
marks int,
primary key (regno,courseno),
foreign key (regno) references student (regno),
foreign key (courseno) references course (courseno)
);
```

```
create table text(
book_isbn int,
book_title varchar(30),
publisher varchar(30),
author varchar(30),
primary key (book_isbn)
);

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)
);
```

QUERY 2

```
use studentenrollment;

insert into student values('1PE11CS002','b','SR','19930924');
insert into student values('1PE11CS003','c','SR','19931127');
insert into student values('1PE11CS004','d','SR','19930413');
insert into student values('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');
insert into course values(112,'EC','CSE');
insert into course values(113,'SS','ISE');
insert into course values (114,'DBMS','CSE');
insert into course values (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(10,'DATABASE SYSTEMS','PEARSON','Schield');
insert into text values(900,'OPERATING SYS','PEARSON','Leland');
insert into text values(901,'CIRCUITS','HALL INDIA','Bob');
insert into text values(902,'SYSTEM SOFTWARE','peterson','Jacob');
insert into text values(903,'SCHEDULING','PEARSON','Patil');
insert into text values(904,'DATABASE SYSTEMS','PEARSON','Jacob');
insert into text values(905,'DATABASE MANAGER','PEARSON','Bob');
insert into text values(906,'SIGNALS','HALL INDIA','Sumit');
select * from text;
```

book_isbn	book_title	publisher	author
10	DATABASE SYSTEMS	PEARSON	Schield
900	OPERATING SYS	PEARSON	Le Schield
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 values('1PE11CS002',114,5,100);
insert into enroll values('1PE11CS003',113,5,100);
insert into enroll values('1PE11CS004',111,5,100);
insert into enroll values('1PE11CS005',112,3,100);
select * from enroll;
```

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
NULL	NULL	NULL	NULL

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

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
NULL	NULL	NULL	NULL

QUERY 3

use studentenrollment;

insert into text values (907,'ai','HALL INDIA','Sumit');
insert into book_adoption values(115, 2, 907);
select * from text;
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
115	2	907
NULL	NULL	NULL

QUERY 4

use studentenrollment;

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

QUERY 5

use studentenrollment;

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*)
Write SQL queries to

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

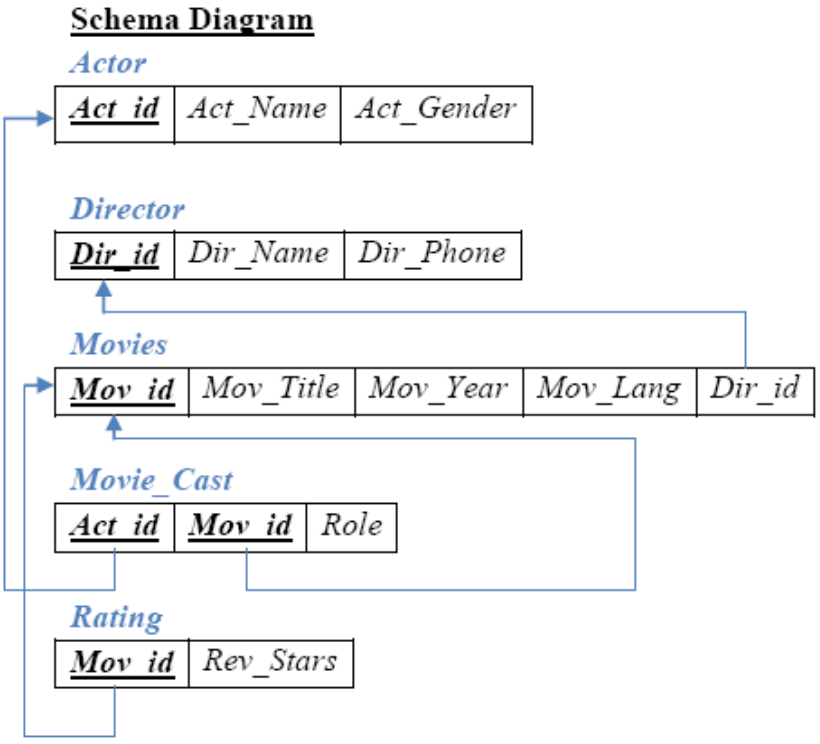
2. Find the movie names where one or more actors acted in two or more movies.

3. List all actors who acted in a movie before 2000 and also in a movie after

2015 (use JOIN operation).

4. 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.

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



QUERY 1

```
create database movie_database;
use movie_database;

create table actor(
act_id int,
act_name varchar(30),
act_gender enum('M','F'),
primary key(act_id));

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

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);

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);

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);
```

QUERY 2

```
use movie_database;
insert into actor values(100, "Leonardo 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

QUERY 3

```
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

QUERY 4

```
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

QUERY 5

```
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

QUERY 6

```
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

QUERY 7

```
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

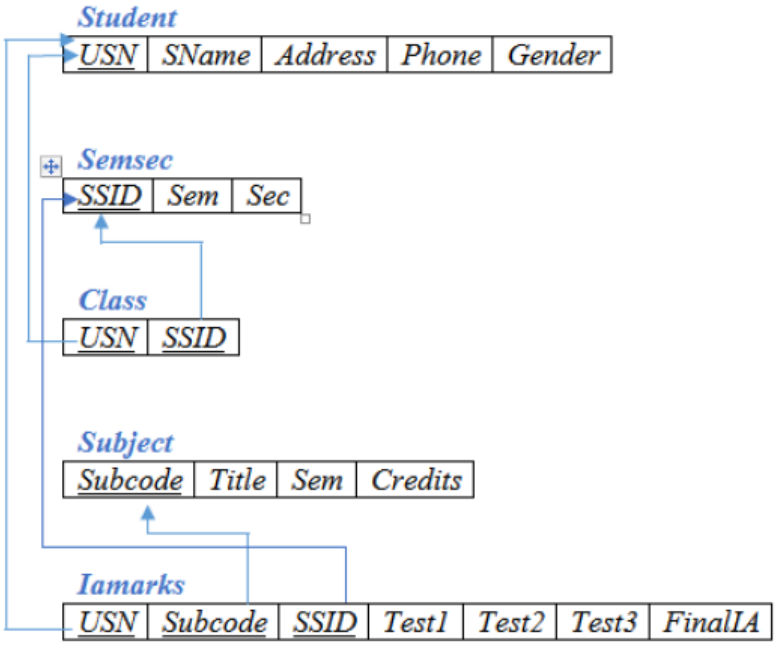
PROGRAM 10: COLLEGE DATABASE

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)
Write SQL queries to

1. List all the student details studying in fourth semester 'C' section.
2. Compute the total number of male and female students in each semester and in each section.
3. Create a view of Test1 marks of student USN '1BI15CS101' in all subjects.
4. 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.

Schema Diagram



QUERY 1

```
create database college_database;
use college_database;

create table student(
usn varchar(30),
sname varchar(30),
address varchar(30),
phone real,
```

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

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

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));

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

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));
```

QUERY 2

```
use college_database;
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
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
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
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

QUERY 3

```
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

QUERY 4

```
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	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

QUERY 5

```
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

QUERY 6

```
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