

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)

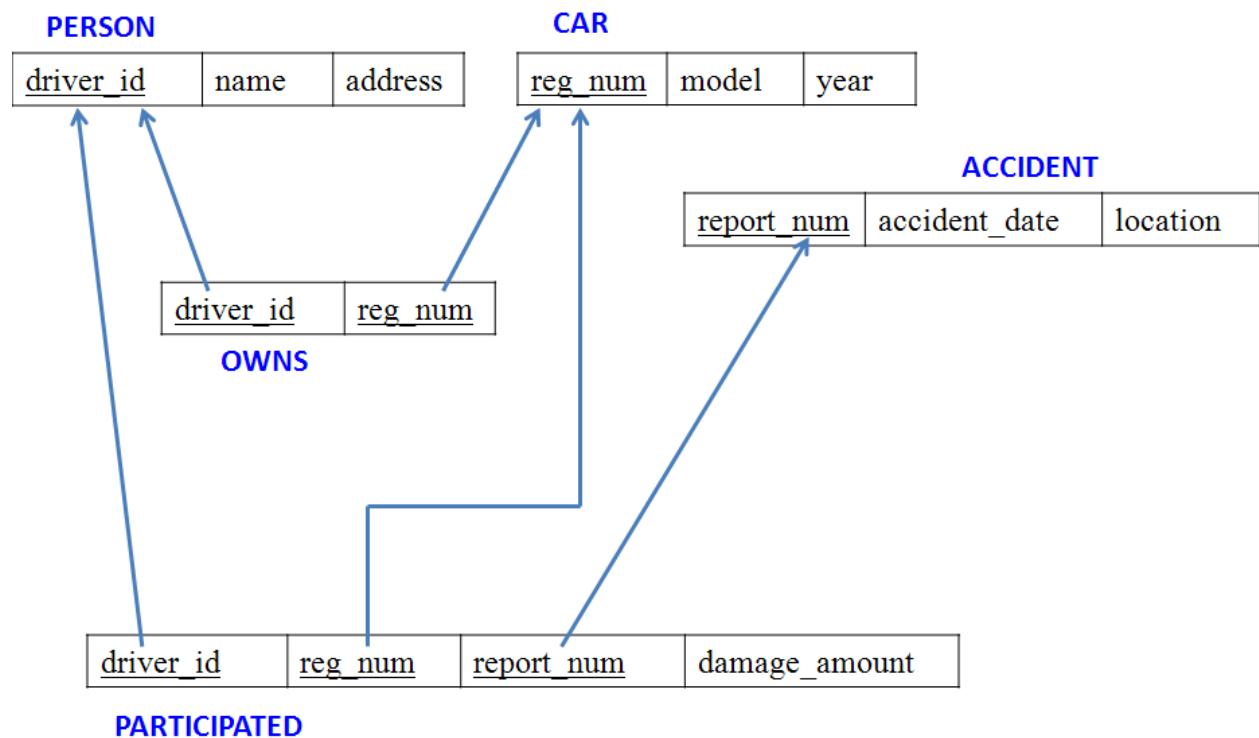
ACCIDENT (report_num: int, accident_date: date, location: String)

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.
- ii) Enter at least five tuples for each relation.
- 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.
 - iv) Find the total number of people who owned cars that involved in accidents in 2008.
 - v) Find the number of accidents in which cars belonging to a specific model (example)were involved.

Schema diagram



Tables

PERSON

driver_id	name	address
A01	Richard	Srinivas nagar
A02	Pradeep	Rajaji nagar
A03	Smith	Ashok nagar
A04	Venu	N R Colony
A05	Jhon	Hanumanth nagar

CAR

reg_num	model	year
KA052250	Indica	1990
KA031181	Lancer	1957
KA095477	Toyota	1998
KA053408	Honda	2008
KA041702	Audi	2005

OWNS

driver_id	reg_num
A01	KA052250
A02	KA053408
A03	KA031181
A04	KA095477
A05	KA041702

ACCIDENT

report_num	accident_date	location
11	01-JAN-03	Mysore Road
12	02-FEB-04	South end Circle
13	21-JAN-03	Bull temple Road
14	17-FEB-08	Mysore Road
15	04-MAR-05	Kanakpura Road

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

```
create database s1;  
use s1;  
  
create table PERSON(  
    driver_id char(20) NOT NULL,  
    Name_ char(30),  
    address char(50),  
    PRIMARY KEY (driver_id)  
);
```

```
create table car(  
    reg_num char(20),  
    model char(30),  
    year int,  
    PRIMARY KEY(reg_num)  
);
```

```
create table ACCIDENT(  
    report_num int,  
    accident_date date,  
    location char(50),  
    PRIMARY KEY(report_num)  
);
```

```
create table OWNS(  
    driver_id char(20),  
    reg_num char(20),  
    FOREIGN KEY(driver_id) references PERSON(driver_id),  
    FOREIGN KEY(reg_num) references car(reg_num)  
);
```

```
create table PARTICIPATED(
    driver_id char(20),
    reg_num char(20),
    report_num int,
    damage_amount int,
    FOREIGN KEY(driver_id) references PERSON(driver_id),
    FOREIGN KEY(reg_num) references car(reg_num),
    FOREIGN KEY(report_num) references ACCIDENT(report_num)
);
```

```
insert into PERSON
values ("A01","RICHARD","SRINIVAS NAGAR"),("A02","PRADEEP","RAJAJI
NAGAR"),("A03","SMITH","ASHOK NAGAR")
,("A04","VENU","N R COLONY"),("A05","JHON","HANUMANYH NAGAR");
```

```
insert into car
values ("KA052250","INDICA",1990),("KA031181","LANCER",1957),("KA095477","TOYOTA",1998)
,("KA053408","HONDA",2008),("KA041702","AUDI",2005);
```

```
insert into ACCIDENT
values (11,"2003-01-01","MYSORE ROAD"),(12,"2004-02-02","SOUTH END CIRCLE"),(13,"2003-01-
21","BULL TEMPLE ROAD");
```

```
insert into ACCIDENT
values (14,"2008-02-17","MYSORE ROAD"),(15,"2005-03-04","KANAKPURA ROAD");
```

```
insert into OWNS
values
("A01","KA052250"),("A02","KA053408"),("A03","KA031181"),("A04","KA095477"),("A05","KA041702");
```

```
insert into PARTICIPATED
values ("A01","KA052250",11,10000),("A02","KA053408",12,50000),("A03","KA095477",13,25000)
,("A04","KA031181",14,3000),("A05","KA041702",15,5000);
select* from PARTICIPATED;
```

```
update PARTICIPATED
SET damage_amount=25000
WHERE reg_num="KA053408";
```

```
insert into ACCIDENT
values (16,"2018-03-29","KORMANGLA");
```

```
SELECT COUNT(accident_date) AS accidentsin2008 FROM ACCIDENT
WHERE YEAR(accident_date)=2008;
```

```
SELECT COUNT(model) AS carwithhondaomodel FROM car
WHERE model="HONDA";
```

```
select* from ACCIDENT
where accident_date="2008-02-17";
```

```
insert into PERSON  
values ("A06","JOHN","BANSANKARI SATGE 2");
```

```
insert into car  
values ("KA05MC001","AUDI",2018);  
select * from car;
```

```
insert into ACCIDENT  
values (17,"2019-03-01","BULL TEMPLE RD");  
select * from ACCIDENT;
```

```
insert into OWNS  
values ("A06","KA05MC001");  
select * from OWNS;
```

```
insert into PARTICIPATED  
values ("A06","KA05MC001",17,75000);
```

```
select* from PERSON;
```

	driver_id	Name_	address
▶	A01	RICHARD	SRINIVAS NAGAR
	A02	PRADEEP	RAJAJI NAGAR
	A03	SMITH	ASHOK NAGAR
	A04	VENU	N R COLONY
	A05	JHON	HANUMANYH NAGAR

```
select* from car;
```

	reg_num	model	year
▶	KA031181	LANCER	1957
	KA041702	AUDI	2005
	KA052250	INDICA	1990
	KA053408	HONDA	2008
	KA095477	TOYOTA	1998

select* from ACCIDENT;

	report_num	accident_date	location
▶	11	2003-01-01	mysore road
	12	2004-02-02	south end circle
	13	2003-01-21	BULL TEMPLE ROAD
	14	2008-02-17	mysore road
	15	2005-03-04	KANAKPURA ROAD
*	16	2018-03-29	KORMANGLA
*	NULL	NULL	NULL

select* from OWNS;

	driver_id	reg_num
▶	A01	KA052250
	A02	KA053408
	A03	KA031181
	A04	KA095477
	A05	KA041702

select * from PARTICIPATED;

	driver_id	reg_num	report_num	damage_amount
▶	A01	KA052250	11	10000
	A02	KA053408	12	25000
	A03	KA095477	13	25000
	A04	KA031181	14	3000
	A05	KA041702	15	5000

SELECT driver_id, reg_num FROM OWNS WHERE reg_num IN(SELECT reg_num FROM car WHERE model="indica");

	driver_id	reg_num
▶	A01	KA052250

```
SELECT reg_num FROM PARTICIPATED WHERE report_num IN(SELECT report_num FROM ACCIDENT WHERE YEAR(accident_date)>=2003 AND YEAR(accident_date)<=2005);
```

	reg_num
▶	KA052250
	KA053408
	KA095477
	KA041702

```
SELECT COUNT(*) FROM ACCIDENT WHERE location LIKE "%MYSORE%";
```

	COUNT(*)
▶	2

```
SELECT driver_id FROM PARTICIPATED WHERE damage_amount>=(SELECT AVG(damage_amount) FROM PARTICIPATED);
```

	driver_id
▶	A02
	A03

```
SELECT COUNT(*) FROM ACCIDENT WHERE YEAR(accident_date)=2008;
```

	COUNT(*)
▶	1

```
DELETE FROM PERSON WHERE Name_="JOHN";
```

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)

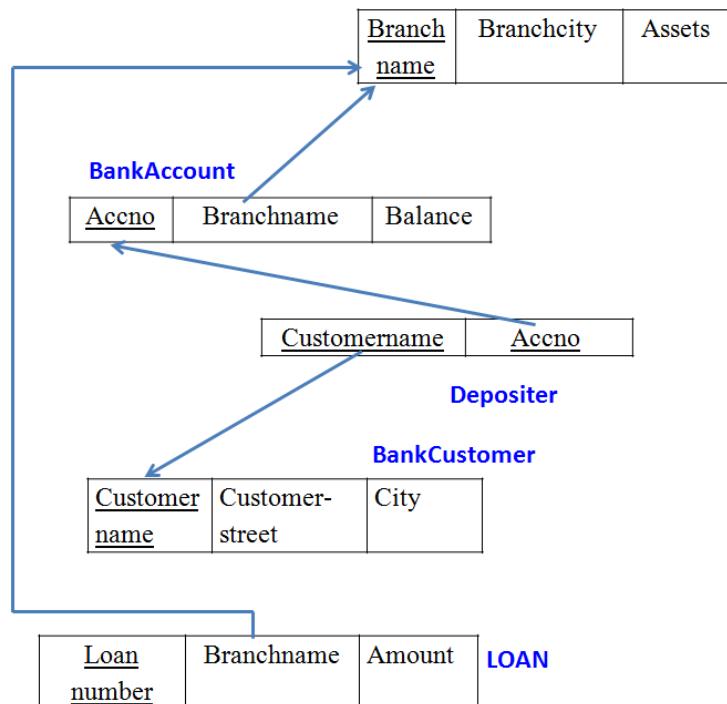
Depositer(customer-name: String, accno: int)

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



Sample Table data

Branch

BRANCHNAME	BRANCHCITY	ASSESTS
SBI_Chamrajpet	Bangalore	50000
SBI_ResidencyRoad	Bangalore	10000
SBI_ShivajiRoad	Bombay	20000
SBI_ParliamentRoad	Delhi	10000
SBI_Jantarmantar	Delhi	20000

BankAccount

ACCNO	BRANCHNAME	BALANCE
1	SBI_Chamrajpet	2000
2	SBI_ResidencyRoad	5000
3	SBI_ShivajiRoad	6000
4	SBI_ParliamentRoad	9000
5	SBI_Jantarmantar	8000
6	SBI_ShivajiRoad	4000
8	SBI_ResidencyRoad	4000
9	SBI_ParliamentRoad	3000
10	SBI_ResidencyRoad	5000
11	SBI_Jantarmantar	2000

BankCustomer

CUSTOMERNAME	CUSTOMERSTREET	CUSTOMERCITY
Avinash	Bull_Temple_Road	Bangalore
Dinesh	Bannergatta_Road	Bangalore
Mohan	NationalCollege_Road	Bangalore
Nikil	Akbar_Road	Delhi
Ravi	Prithviraj_Road	Delhi

Loan

LOANNUMBER	BRANCHNAME	AMOUNT
1	SBI_Chamrajpet	1000
2	SBI_ResidencyRoad	2000
3	SBI_ShivajiRoad	3000
4	SBI_ParliamentRoad	4000
5	SBI_Jantarmantar	5000

Depositer

CUSTOMERNAME	ACCNO
Avinash	1
Dinesh	2
Nikil	4
Ravi	5
Avinash	8
Nikil	9
Dinesh	10
Nikil	11

```
create database sample11;
```

```
use sample11;
```

```
CREATE TABLE branch
```

```
( branch_name VARCHAR(20),
branch_city VARCHAR(20),
assets REAL,
PRIMARY KEY(branch_name)
);
```

```
CREATE TABLE accounts
```

```
( acc_no INT,  
branch_name VARCHAR(50),  
balance REAL,  
PRIMARY KEY(acc_no),  
FOREIGN KEY(branch_name) REFERENCES branch(branch_name)  
ON UPDATE CASCADE ON DELETE CASCADE  
);
```

```
CREATE TABLE customer  
( customer_name VARCHAR(20),  
customer_street VARCHAR(50),  
customer_city VARCHAR(20),  
PRIMARY KEY(customer_name)  
);
```

```
CREATE TABLE depositor  
( customer_name VARCHAR(20),  
acc_no INT,  
PRIMARY KEY(customer_name, acc_no),  
FOREIGN KEY(customer_name) REFERENCES customer(customer_name)  
ON UPDATE CASCADE ON DELETE CASCADE,  
FOREIGN KEY(acc_no) REFERENCES accounts(acc_no)  
ON UPDATE CASCADE ON DELETE CASCADE  
);
```

```
CREATE TABLE loan  
( loan_number INT,  
branch_name VARCHAR(50),  
amount REAL,
```

```
PRIMARY KEY(loan_number),
FOREIGN KEY(branch_name) REFERENCES branch(branch_name)
ON UPDATE CASCADE ON DELETE CASCADE
);
```

```
INSERT INTO branch(branch_name,branch_city,assets) VALUES
('SBI_Chamrajpet','Bangalore',50000),('SBI_ResidencyRoad','Bangalore',10000),('SBI_ShivajiRoad','Bombay',20000),('SBI_ParliamentRoad','Delhi',10000),('SBI_Jantarmantar','Delhi',20000);
```

```
INSERT INTO accounts(acc_no,branch_name,balance) VALUES
(1,'SBI_Chamrajpet',2000),(2,'SBI_ResidencyRoad',5000),(3,'SBI_ShivajiRoad',6000),(4,'SBI_ParliamentRoad',9000),(5,'SBI_Jantarmantar',8000),(6,'SBI_ShivajiRoad',4000),(8,'SBI_ResidencyRoad',4000),(9,'SBI_ParliamentRoad',3000),(10,'SBI_ResidencyRoad',5000),(11,'SBI_Jantarmantar',2000);
```

```
INSERT INTO customer(customer_name,customer_street,customer_city) VALUES
('Avinash','Bull_Temple_Road','Bangalore'),('Dinesh','Bannerghatta_Road','Bangalore'),('Mohan','National College_Road','Bangalore'),('Nikil','Akbar_Road','Delhi'),('Ravi','Prithviraj_Road','Delhi');
```

```
INSERT INTO depositor(customer_name,acc_no) VALUES
('Avinash',1),('Dinesh',2),('Nikil',4),('Ravi',5),('Avinash',8),('Nikil',9),('Dinesh',10),('Nikil',11);
```

```
INSERT INTO loan(loan_number,branch_name,amount) VALUES
(1,'SBI_Chamrajpet',1000),(2,'SBI_ResidencyRoad',2000),(3,'SBI_ShivajiRoad',3000),(4,'SBI_ParliamentRoad',4000),(5,'SBI_Jantarmantar',5000);
```

```
SELECT * FROM branch;
```

Result Grid | Filter Rows: Edit: Export

	branch_name	branch_city	assets
▶	SBI_Chamrajpet	Bangalore	50000
	SBI_Jantarmantar	Delhi	20000
	SBI_ParliamentRoad	Delhi	10000
	SBI_ResidencyRoad	Bangalore	10000
	SBI_ShivajiRoad	Bombay	20000
●	NULL	NULL	NULL

SELECT * FROM accounts;

Result Grid | Filter Rows: Edit: Export/Import

	acc_no	branch_name	balance
▶	1	SBI_Chamrajpet	2000
	2	SBI_ResidencyRoad	5000
	4	SBI_ParliamentRoad	9000
	5	SBI_Jantarmantar	8000
	8	SBI_ResidencyRoad	4000
	9	SBI_ParliamentRoad	3000
	10	SBI_ResidencyRoad	5000
	11	SBI_Jantarmantar	2000
●	NULL	NULL	NULL

SELECT * FROM customer;

Result Grid | Filter Rows: Edit: Export/Import

	customer_name	customer_street	customer_city
▶	Avinash	Bull_Temple_Road	Bangalore
	Dinesh	Bannergatta_Road	Bangalore
	Mohan	NationalCollege_Road	Bangalore
	Nikil	Akbar_Road	Delhi
	Ravi	Prithviraj_Road	Delhi
●	NULL	NULL	NULL

```
SELECT * FROM depositor;
```

Result Grid | Filter Rows: | Edit: | Export

	customer_name	acc_no
▶	Avinash	1
	Dinesh	2
	Nikil	4
	Ravi	5
	Avinash	8
	Nikil	9
	Dinesh	10
	Nikil	11
*	HULL	HULL

```
SELECT * FROM loan;
```

Result Grid | Filter Rows: | Edit: | Export

	loan_number	branch_name	amount
▶	1	SBI_Chamrajpet	1000
	2	SBI_ResidencyRoad	2000
	3	SBI_ShivajiRoad	3000
	4	SBI_ParliamentRoad	4000
	5	SBI_Jantarmantar	5000
*	HULL	HULL	HULL

```
SELECT * FROM customer WHERE customer_name IN(SELECT customer_name FROM depositor group by customer_name having COUNT(customer_name)>=2);
```

Result Grid | Filter Rows: Edit: Export

	customer_name	customer_street	customer_city
▶	Avinash	Bull_Temple_Road	Bangalore
	Dinesh	Bannergatta_Road	Bangalore
	Nikil	Akbar_Road	Delhi
*	NULL	NULL	NULL

```
SELECT d.customer_name FROM accounts a, depositor d,branch b WHERE d.acc_no=a.acc_no AND
b.branch_name=a.branch_name AND b.branch_city="Delhi" GROUP BY d.customer_name having
count(distinct b.branch_name)=(SELECT COUNT(branch_name) FROM branch WHERE
branch_city="Delhi");
```

Result Grid | Filter Rows: Export

	customer_name
▶	Nikil

```
DELETE FROM ACCOUNTS WHERE branch_name IN(SELECT branch_name FROM BRANCH WHERE
branch_city='Bombay');
```

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:

- i) Find the pnames of parts for which there is some supplier.
- ii) Find the snames of suppliers who supply every part.
- iii) Find the snames of suppliers who supply every red part.
- iv) Find the pnames of parts supplied by Acme Widget Suppliers and by no one else.
- 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).
- vi) For each part, find the sname of the supplier who charges the most for that part.

Schema Diagram

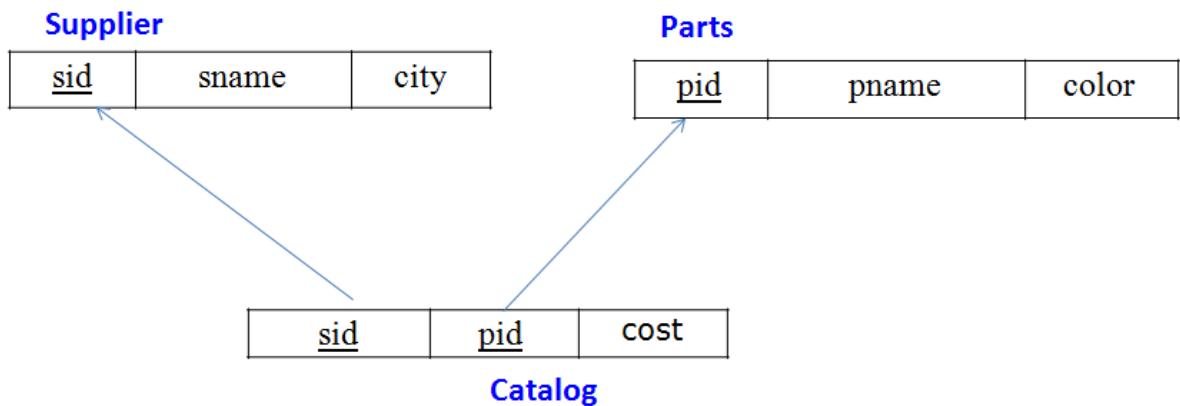


Table Data

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

```

create database supplier;
use supplier;
CREATE TABLE suppliers(
    sid INT,
    sname VARCHAR(20),
    address VARCHAR(50),
    PRIMARY KEY (sid)
);
CREATE TABLE parts(
    pid INT,
    pname VARCHAR(20),
    color VARCHAR(10),
    PRIMARY KEY (pid)
);
CREATE TABLE catalog(
    sid INT,
    pid INT,
    cost REAL,
    PRIMARY KEY(sid,pid),

```

```

FOREIGN KEY(sid) REFERENCES suppliers(sid)
ON delete CASCADE ON update CASCADE,
FOREIGN KEY(pid) REFERENCES parts(pid)
ON delete CASCADE ON update CASCADE
);

```

```

insert into suppliers values (10001,'Acme Widget','Bangalore'), (10002,'Johns','Kolkata'),
(10003,'Vimal','Mumbai'),(10004,'Reliance','Delhi');
insert into parts values
(20001,'Book','Red'),(20002,'Pen','Red'),(20003,'Pencil','Green'),(20004,'Mobile','Green'),(20005,
'Charger','Black');
insert into catalog
values(10001,20001,10),(10001,20002,10),(10001,20003,30),(10001,20004,10),(10001,20005,
10),(10002,20001,10),(10002,20002,20),(10003,20003,30),(10004,20003,40);

```

```
SELECT * FROM suppliers;
```

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

```
SELECT * FROM parts;
```

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

```
SELECT * FROM catalog;
```

The screenshot shows the MySQL Workbench interface with the 'Result Grid' tab selected. The query 'SELECT * FROM catalog;' has been run, and the results are displayed in a table. The table has three columns: sid, pid, and cost. The data consists of ten rows, with the last row being a blank row indicated by asterisks (*).

	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

```
SELECT DISTINCT p.pname FROM parts p, catalog c WHERE p.pid = c.pid;
```

The screenshot shows the MySQL Workbench interface with the 'Result Grid' tab selected. The query 'SELECT DISTINCT p.pname FROM parts p, catalog c WHERE p.pid = c.pid;' has been run, and the results are displayed in a table. The table has one column: pname. The data consists of five distinct product names: Book, Pen, Pencil, Mobile, and Charger.

	pname
▶	Book
	Pen
	Pencil
	Mobile
	Charger

```
select suppliers.sname from suppliers where suppliers.sid in(select catalog.sid from catalog  
inner join  
parts on catalog.pid=parts.pid group by catalog.sid having count(*)=(select count(parts.pid) from  
parts));
```

The screenshot shows the MySQL Workbench interface with the 'Result Grid' tab selected. The query 'select suppliers.sname from suppliers where suppliers.sid in(select catalog.sid from catalog inner join parts on catalog.pid=parts.pid group by catalog.sid having count(*)=(select count(parts.pid) from parts));' has been run, and the results are displayed in a table. The table has one column: sname. The data consists of a single row with the value 'Acme Widget'.

	sname
▶	Acme Widget

```

select suppliers.sname from suppliers where suppliers.sid in (select catalog.sid from catalog
inner join
parts on catalog.pid=parts.pid where catalog.pid in (select parts.pid from parts where
parts.color='Red') group by catalog.sid having count(*)=(select count(parts.color) from parts
where
parts.color='Red'));

```

Result Grid			Filter Rows:	Export:
sname				
▶ Acme Widget				
Johns				

```

SELECT p.pname FROM parts p, catalog c, suppliers s WHERE p.pid = c.pid AND c.sid = s.sid
AND s.sname = 'Acme Widget' AND NOT EXISTS ( SELECT * FROM catalog c1, suppliers s1
WHERE p.pid = c1.pid AND c1.sid = s1.sid AND s1.sname <> 'Acme Widget');

```

Result Grid			Filter Rows:	Export:
pname				
▶ Mobile				
Charger				

```

SELECT DISTINCT c.sid FROM catalog c WHERE c.cost > (SELECT AVG(C1.cost) FROM
catalog c1 WHERE c1.pid = c.pid) ;

```

Result Grid			Filter Rows:	Export:
sid				
▶ 10002				
10004				

```

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);

```

| Result Grid | Filter Rows: | Export: |

	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, meetsat: 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.

- i. Find the names of all Juniors (level = JR) who are enrolled in a class taught by “name”
- ii. Find the names of all classes that either meet in room R128 or have five or more Students enrolled.
- iii. Find the names of all students who are enrolled in two classes that meet at the same time.
- iv. Find the names of faculty members who teach in every room in which some class is taught.
- v. Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.

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

SQL> select * from student;

SNUM	SNAME	MA	LV	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

SQL> select * from faculty;

FID	FNAME	DEPTID
11	Harish	1000
12	MV	1000
13	Mira	1001
14	Shiva	1002
15	Nupur	1000

SQL> select * from class;

CNAME	METTS_A	ROOM	FID
Class1	12/11/15 10:15:16.00000	R1	14
Class10	12/11/15 10:15:16.00000	R128	14
Class2	12/11/15 10:15:20.000000	R2	12
Class3	12/11/15 10:15:25.000000	R3	11
Class4	12/11/15 20:15:20.000000	R4	14
Class5	12/11/15 20:15:20.000000	R3	15
Class6	12/11/15 13:20:20.000000	R2	14
Class7	12/11/15 10:10:10.000000	R3	14

SQL> select * from enrolled;

SNUM	CNAME
1	class1
2	class1
3	class3
4	class3
5	class4

Program:-

```
create database studentfaculty4;
use studentfaculty4;

create table STUDENT(
    snum int,
    sname varchar(60),
    major varchar(50),
    lvl varchar(50),
    age int,
    primary key(snum)
);

create table CLASS(
    cname varchar(60),
    meetsat timestamp,
    room varchar(60),
    fid int,
    primary key (cname)
);

create table enrolled(
    snum int,
    cname varchar(60),
    primary key(snum,cname),
    foreign key(snum) references STUDENT(snum)
        on update cascade on delete cascade,
    foreign key(cname) references CLASS(cname)
```

```
on update cascade on delete cascade  
);
```

```
create table FACULTY(  
fid int,  
fname varchar(60),  
deptid int,  
primary key(fid)  
);
```

```
insert into STUDENT values (1,'Jhon','CS','Jr',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);
```

```
update STUDENT set lvl='Sr' where sname='Jhon';
```

```
insert into CLASS values ('Class1',"12/11/15 10:15:16.00000",'R1',14);
```

```
select * from CLASS;
```

```
delete from CLASS where cname='Class1';
```

```
select * from CLASS;
```

```
insert into CLASS values ('Class1',"15/11/12 10:15:16.00000",'R1',14);
```

```
select * from CLASS;
```

```
insert into CLASS values ('Class10',"15/11/12 10:15:16.00000",'R128',14), ('Class2',"15/11/12  
10:15:20.00000",'R2',12),
```

```
('Class3',"15/11/12 10:15:25.00000",'R3',11), ('Class4',"15/11/12 10:15:20.00000",'R4',14),  
('Class5',"15/11/12 10:15:20.00000",'R3',15),
```

```
('Class6',"15/11/12 13:20:20.00000",'R2',14), ('Class7',"15/11/12 10:10:10.00000",'R3',14);
```

```
insert into ENROLLED values (1,'Class1'),(2,'Class1'),(3,'Class3'),(4,'Class3'),(5,'Class4');
```

```
insert into FACULTY values  
(11,'Harish',1000),(12,'MV',1000),(13,'Mira',1001),(14,'Shiva',1002),(15,'Nupur',1000);
```

select * from STUDENT;

	snum	sname	major	lv	age
▶	1	Jhon	CS	Jr	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

select * from CLASS;

	cname	meetsat	room	fid
▶	Class1	2015-11-12 10:15:16	R1	14
	Class10	2015-11-12 10:15:16	R128	14
	Class2	2015-11-12 10:15:20	R2	12
	Class3	2015-11-12 10:15:25	R3	11
	Class4	2015-11-12 10:15:20	R4	14
	Class5	2015-11-12 10:15:20	R3	15
	Class6	2015-11-12 13:20:20	R2	14
	Class7	2015-11-12 10:10:10	R3	14
	NULL	NULL	NULL	NULL

select * from FACULTY;

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

select * from ENROLLED;

	snum	cname
▶	1	Class1
	2	Class1
	3	Class3
	4	Class3
*	5	Class4
	NULL	NULL

```
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 = 'Shiva' AND s.lvl = 'Jr';
```

	sname
▶	Jhon
	Smith
	Rahul

```
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(e.snum)>4);
```

	cname
▶	Class10
*	NULL

```
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
--	-------

```
create table ref1 as  
select count(distinct(CLASS.room))as c from CLASS where CLASS cname in  
(select distinct(ENROLLED.cname) from ENROLLED)  
;  
select FACULTY.fname from FACULTY where FACULTY.fid in (select (CLASS.fid)  
from CLASS where class.room in( select distinct(class.room) from class  
where CLASS cname in (select distinct(ENROLLED.cname) from ENROLLED))  
group by CLASS.fid having count(*)=(select ref1.c from ref1));
```

	fname
--	-------

▶	Shiva
---	-------

```
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
	Nupur

```
SELECT DISTINCT s.sname FROM STUDENT s WHERE s.snum NOT IN(SELECT e.snum FROM ENROLLED e);
```

	sname
▶	Rita

```
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.age, s1.lvl HAVING COUNT(*) >= ALL (SELECT COUNT(*) FROM STUDENT s2 WHERE s1.age=s2.age GROUP BY s2.lvl, s2.age)) ORDER BY s.age;
```

	age	lvl
▶	19	Jr
	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.

- i.Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80,000.
- 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.
- iii.Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Frankfurt.
- 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.
- v.Find the names of pilots certified for some Boeing aircraft.
- vi.Find the aids of all aircraft that can be used on routes from Bengaluru to New Delhi.
- 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.

SQL> select * from Flights;

FLNO	FFROM	TO	DISTANCE	DEPARTS	ARRIVES	PRICE
101	Bangalore	Delhi	2500	13-MAY-05 07:15:31.000000 AM	13-MAY-05 07:15:31.000000 A	5000
102	Bangalore	Lucknow	3000	05/05/13 07:15:31.000000	05/05/13 11:15:31.000000	6000
103	Lucknow	Delhi	500	05/05/13 12:15:31.000000	05/05/13 17:15:31.000000	3000
107	Bangalore	Frankfurt	8000	05/05/13 07:15:31.000000	05/05/13 22:15:31.000000	60000
104	Bangalore	Frankfurt	8500	05/05/13 07:15:31.000000	05/05/13 23:15:31.000000	75000
105	Kolkata	Delhi	3400	05/05/13 07:15:31.000000	05/05/13 09:15:31.000000	7000

SQL> 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

7 rows selected.

SQL> 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

EID	AID
703	105
704	105
705	103

14 rows selected.

SQL> 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

7 rows selected.

Program :-

```
CREATE DATABASE AIRLINE_FLIGHT_DATABASE;
```

```
USE AIRLINE_FLIGHT_DATABASE;
```

```
CREATE TABLE FLIGHTS
```

```
(
```

```
    flno int,
```

```
    ffrom varchar(40),
```

```
    tto varchar(40),
```

```
    distance int,
```

```
    departs datetime,
```

```
    arrives datetime,
```

```
    price int,
```

```
    primary key(flno)
```

```
);
```

```
CREATE TABLE AIRCRAFT
```

```
(
```

```
    aid int,
```

```
    aname varchar(40),
```

```
    cruisingrange int,
```

```
    primary key(aid)
```

```
);
```

```
CREATE TABLE EMPLOYEES
```

```
(
```

```
    eid int,
```

```
ename varchar(40),  
salary int,  
primary key(eid)  
);
```

```
CREATE TABLE CERTIFIED  
(  
    eid int,  
    aid int,  
    FOREIGN KEY(aid) REFERENCES AIRCRAFT(aid),  
    FOREIGN KEY(eid) REFERENCES EMPLOYEES(eid)  
);
```

INSERT INTO FLIGHTS

```
VALUES (101,"Bangalore","Delhi",2500,'2005-05-13:07:15:31.000000','2005-05-  
13:07:15:31.000000',5000),  
(102,"Bangalore","Lucknow",3000,'2013-05-05:07:15:31.000000','2013-05-  
05:11:15:31.000000',6000),  
(103,"Lucknow","Delhi",500,'2013-05-05:12:15:31.000000','2013-05-05:17:15:31.000000',3000),  
(107,"Bangalore","Frankfurt",8000,'2013-05-05:07:15:31.000000','2013-05-  
05:22:15:31.000000',60000),  
(104,"Bangalore","Frankfurt",8500,'2013-05-05:07:15:31.000000','2013-05-  
05:23:15:31.000000',75000),  
(105,"Kolkata","Delhi",3400,'2013-05-05:07:15:31.000000','2013-05-05:09:15:31.000000',7000);  
SELECT * FROM FLIGHTS;
```

	fno	ffrom	tto	distance	departs	arrives	price
▶	101	Bangalore	Delhi	2500	2005-05-13 07:15:31	2005-05-13 07:15:31	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
	107	Bangalore	Frankfurt	8000	2013-05-05 07:15:31	2013-05-05 22:15:31	60000
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

INSERT INTO AIRCRAFT

```
VALUES (101,747,3000),(102,"Boeing",900),(103,647,800),(104,"Dreamliner",10000),
(105,"Boeing",3500),(106,707,1500),(107,"Dream",120000);
```

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),(702,"B",100000),(703,"C",150000),(704,"D",90000),
(705,"E",40000),(706,"F",60000),(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),(701,102),(701,106),(701,105),(702,104),(703,104),(704,104),(702,107),
       (703,107),(704,107),(702,101),(703,105),(704,105),(705,103);
```

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
	703	105
	704	105
	705	103

```
SELECT distinct a.aname
FROM AIRCRAFT a,EMPLOYEES e,CERTIFIED c
WHERE a.aid=c.aid and e.eid=c.eid and e.salary>80000;
```

	aname
▶	Dreamliner
	Dream
	747
	Boeing

```

SELECT e.eid,e.ename,max(a. cruisingrange)
FROM EMPLOYEES e,CERTIFIED c,AIRCRAFT a
WHERE e.eid=c.eid and a.aid=c.aid
group by e.ename
having count(c.aid)>3;

```

	eid	ename	max(a. cruisingrange)
▶	701	A	3500

```

SELECT e.ename
FROM EMPLOYEES e
WHERE salary < (select min(price)
from FLIGHTS
where ffrom="Bangalore" and tto="Frankfurt");

```

	ename
▶	A
	E

```
SELECT a.aname,a.cruisingrange,avg(e.salary)
```

```

FROM AIRCRAFT a,EMPLOYEES e,CERTIFIED c
WHERE c.eid=e.eid and c.aid=a.aid
group by a.ename
having a.cruisingrange > 1000;

```

	ename	cruisingrange	avg(e.salary)
▶	747	3000	75000.0000
	Dreamliner	10000	113333.3333
	707	1500	50000.0000
	Dream	120000	113333.3333

```

SELECT distinct e.ename
FROM EMPLOYEES e,CERTIFIED c,AIRCRAFT a
WHERE e.eid=c.eid and a.aid=c.aid and aname like "Boeing";

```

	ename
▶	A
	C
	D

```

SELECT a.aid
FROM AIRCRAFT a
WHERE a. cruisingrange >= (select distance
                           from FLIGHTS
                           where ffrom="Bangalore" and tto="Delhi");

```

	aid
▶	101
	104
	105
	107
*	NULL

```
SELECT f.ffrom,f.tto,f.arrives
FROM FLIGHTS f
WHERE (f.ffrom="Bangalore" and f.tto=(select ffrom
                                         from FLIGHTS
where tto="Kolkata")) or f.tto="Kolkata";
```

	ffrom	tto	arrives

Program 6 : Order Database

Consider the following schema for Order Database:

SALESMAN (*Salesman_id, Name, City, Commission*)

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

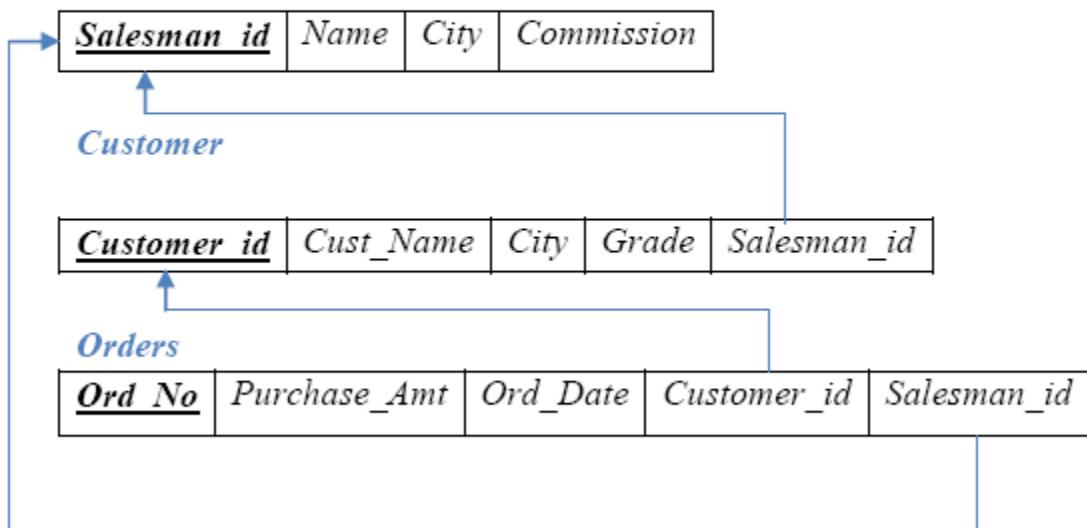
ORDERS (*Ord_No, Purchase_Amt, Ord_Date, Customer_id, Salesman_id*)

Write SQL queries to

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.

Schema Diagram

Salesman



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 %

SELECT * FROM CUSTOMER1;

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

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
52	1000	24-FEB-17	13	2000
53	3500	13-APR-17	14	3000
54	550	09-MAR-17	12	2000

Program:-

```

create database orderdb2;
use orderdb2;
create table salesman(
salesman_id varchar(20),
salesman_name varchar(20),
salesman_city varchar(20),
commission varchar(20),
primary key(salesman_id)
);
create table customer(
customer_id varchar(20),
customer_name varchar(20),
customer_city varchar(20),
grade varchar(20),
salesman_id varchar(20),

```

```

primary key(customer_id),
foreign key(salesman_id) references salesman(salesman_id) on delete set null);
create table orders(
ord_no int,
purchase_amt double,
ord_date date,
customer_id varchar(20),
salesman_id varchar(20),
foreign key(salesman_id) references salesman(salesman_id) on delete cascade,
foreign key(customer_id) references customer(customer_id) on delete cascade
);
insert into salesman values("1000","JHON","BANGLORE","25%"),
("2000","RAVI","BANGLORE","20%"),
("3000","KUMAR","MYSORE","15%"),
("4000","SMITH","DELHI","30%"),
("5000","HARSHA","HYDRABAD","15%");
select * from salesman;

```

	salesman_id	salesman_name	salesman_city	commission
▶	1000	JHON	BANGLORE	25%
	2000	RAVI	BANGLORE	20%
	3000	KUMAR	MYSORE	15%
	4000	SMITH	DELHI	30%
*	5000	HARSHA	HYDRABAD	15%
	NULL	NULL	NULL	NULL

```

insert into customer values("10","PREETHI","BANGLORE","100","1000"),
("11","VIVEK","MANGLORE","300","1000"),
("12","BHASKAR","CHENNAI","400","2000"),
("13","CHETHAN","BANGLORE","200","2000"),
("14","MAMTHA","BANGLORE","400","3000");
select * from customer;

```

	customer_id	customer_name	customer_city	grade	salesman_id
▶	10	PREETHI	BANGLORE	100	1000
	11	VIVEK	MANGLORE	300	1000
	12	BHASKAR	CHENNAI	400	2000
	13	CHETHAN	BANGLORE	200	2000
*	14	MAMTHA	BANGLORE	400	3000
	NULL	NULL	NULL	NULL	NULL

```

insert into orders values("50","5000","17-05-04","10","1000"),
("51","450","17-01-20","10","2000"),
("52","1000","17-02-24","13","2000"),
("53","3500","17-04-13","14","3000"),
("54","550","17-03-09","12","2000");

```

```
select * from orders;
```

	ord_no	purchase_amt	ord_date	customer_id	salesman_id
▶	50	5000	2017-05-04	10	1000
	51	450	2017-01-20	10	2000
	52	1000	2017-02-24	13	2000
	53	3500	2017-04-13	14	3000
	54	550	2017-03-09	12	2000

```
select grade,count(distinct customer_id) from customer group by grade having grade > (select avg(grade) from customer where customer_city ="BANGLORE");
```

	grade	count(distinct customer_id)
▶	300	1
	400	2

```
select salesman_id ,salesman_name from salesman S where 1 <(select count(*) from customer  
where salesman_id = S.salesman_id);
```

	salesman_id	salesman_name
▶	1000	JHON
	2000	RAVI
*	NUL	NUL

```
select salesman.salesman_id ,salesman_name, customer_name,commission from  
salesman,customer where salesman_city = customer_city union select  
salesman_id,salesman_name , 'NO MATCH FOUND',commission from salesman where not  
salesman_city = any(select customer_city from customer)order by 2 desc;
```

	salesman_id	salesman_name	customer_name	commission
▶	4000	SMITH	NO MATCH FOUND	30%
	2000	RAVI	PREETHI	20%
	2000	RAVI	CHETHAN	20%
	2000	RAVI	MAMTHA	20%
	3000	KUMAR	NO MATCH FOUND	15%
	1000	JHON	PREETHI	25%
	1000	JHON	CHETHAN	25%
	1000	JHON	MAMTHA	25%
	5000	HARSHA	NO MATCH FOUND	15%

```
create view best_salesman as select b.ord_date ,a.salesman_id,a.salesman_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 best_salesman;
```

	ord_date	salesman_id	salesman_name
▶	2017-05-04	1000	JHON
	2017-01-20	2000	RAVI
	2017-02-24	2000	RAVI
	2017-04-13	3000	KUMAR
	2017-03-09	2000	RAVI

```
delete from salesman where salesman_id = 1000;
```

Program 7 : Book Database

BOOK (Book_id, Title, Publisher_Name, Pub_Year)

BOOK_AUTHORS (Book_id, Author_Name)

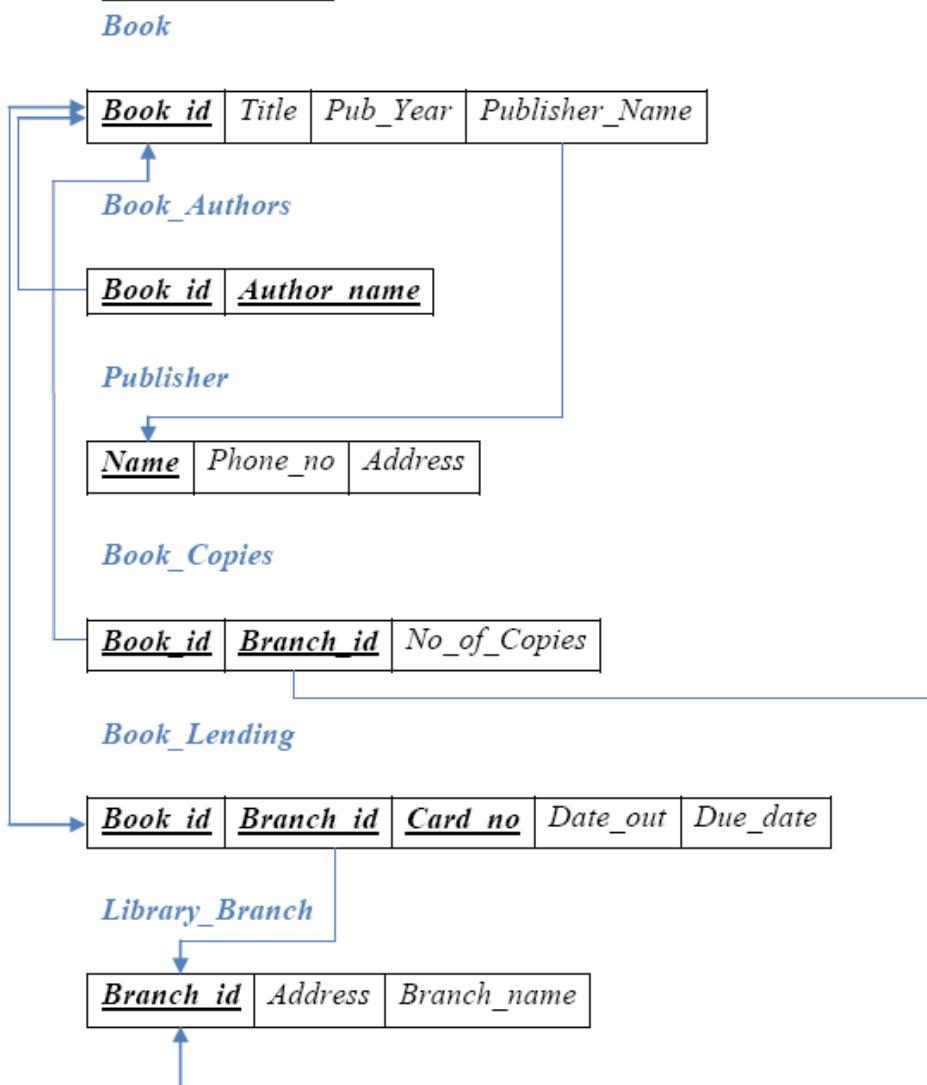
PUBLISHER (Name, Address, Phone)

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



Write SQL queries to

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.

```
SQL> select * from publisher;
```

NAME	PHONE	ADDRESS
MCGRAW-HILL	9989076587	BANGALORE
PEARSON	9889076565	NEWDELHI
RANDOM HOUSE	7455679345	HYDRABAD
HACHETTE LIVRE	8970862340	CHENAI
GRUPO PLANETA	7756120238	BANGALORE

```
SQL> SELECT * FROM BOOK;
```

BOOK_ID	TITLE	PUB_YEAR	PUBLISHER_NAME
1	DBMS	JAN-2017	MCGRAW-HILL
2	ADBMS	JUN-2016	MCGRAW-HILL
3	CN	SEP-2016	PEARSON
4	CG	SEP-2015	GRUPO PLANETA
5	OS	MAY-2016	PEARSON

```
SQL> SELECT * FROM BOOK_AUTHORS;
```

AUTHOR_NAME	BOOK_ID
NAVATHE	1
NAVATHE	2
TANENBAUM	3
EDWARD ANGEL	4
GALVIN	5

```
SQL> 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

```
SQL> 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
1	5	10
3	4	11

```
SQL> select * from book_lending;
```

DATE_OUT	DUE_DATE	BOOK_ID	BRANCH_ID	CARD_NO
01-JAN-17	01-JUN-17	1	10	101
11-JAN-17	11-MAR-17	3	14	101
21-FEB-17	21-APR-17	2	13	101
15-MAR-17	15-JUL-17	4	11	101
12-APR-17	12-MAY-17	1	11	104

```
SQL> SELECT * FROM CARD;
```

CARD_NO
100
101
102
103
104

PROGRAM:-

```
create database book2;
```

```
use book2;
```

```
CREATE TABLE PUBLISHER
```

```
(NAME VARCHAR(20),
```

```
PHONE VARCHAR(11),
```

```
ADDRESS VARCHAR(20),
```

```
primary key(NAME));
```

```
CREATE TABLE BOOK
```

```
(BOOK_ID INT,
```

```
TITLE VARCHAR(20),  
PUB_YEAR VARCHAR(20),  
PUBLISHER_NAME VARCHAR(20),  
PRIMARY KEY(BOOK_ID),  
FOREIGN KEY(PUBLISHER_NAME) REFERENCES PUBLISHER(NAME) ON DELETE CASCADE  
);  
  
CREATE TABLE BOOK_AUTHORS  
(AUTHOR_NAME VARCHAR(20),  
BOOK_ID INT,  
PRIMARY KEY (BOOK_ID, AUTHOR_NAME),  
FOREIGN KEY(BOOK_ID) REFERENCES BOOK(BOOK_ID) ON DELETE CASCADE  
);  
  
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,  
PRIMARY KEY (BOOK_ID, BRANCH_ID),  
FOREIGN KEY(BOOK_ID) REFERENCES BOOK(BOOK_ID) ON DELETE CASCADE,  
FOREIGN KEY(BRANCH_ID) REFERENCES LIBRARY_BRANCH(BRANCH_ID) ON DELETE CASCADE  
);  
  
CREATE TABLE CARD  
(CARD_NO INT PRIMARY KEY);  
  
CREATE TABLE BOOK_LENDING  
(DATE_OUT DATE,
```

```
DUE_DATE DATE,  
BOOK_ID INT,  
BRANCH_ID INT,  
CARD_NO INT,  
PRIMARY KEY (BOOK_ID, BRANCH_ID, CARD_NO),  
FOREIGN KEY(BOOK_ID) REFERENCES BOOK(BOOK_ID) ON DELETE CASCADE,  
FOREIGN KEY(BRANCH_ID) REFERENCES LIBRARY_BRANCH(BRANCH_ID) ON DELETE CASCADE);  
INSERT INTO PUBLISHER VALUES ('MCGRAW-HILL', '9989076587', 'BANGALORE'), ('PEARSON',  
'9889076565', 'NEWDELHI'),  
('RANDOM HOUSE', '7455679345', 'HYDRABAD'), ('HACHETTE LIVRE', '8970862340', 'CHENAI'), ('GRUPO  
PLANETA', '7756120238', 'BANGALORE');  
INSERT INTO BOOK VALUES (1, 'DBMS', 'JAN-2017', 'MCGRAW-HILL'), (2, 'ADBMS', 'JUN-2016', 'MCGRAW-  
HILL'), (3, 'CN', 'SEP-2016', 'PEARSON'),  
(4, 'CG', 'SEP-2015', 'GRUPO PLANETA'), (5, 'OS', 'MAY-2016', 'PEARSON');  
INSERT INTO BOOK_AUTHORS VALUES ('NAVATHE', 1), ('NAVATHE', 2), ('TANENBAUM', 3), ('EDWARD  
ANGEL', 4), ('GALVIN', 5);  
INSERT INTO LIBRARY_BRANCH VALUES (10, 'RR NAGAR', 'BANGALORE'),  
(11, 'RNSIT', 'BANGALORE'), (12, 'RAJAJI NAGAR',  
'BANGALORE'), (13, 'NITTE', 'MANGALORE'), (14, 'MANIPAL', 'UDUPI');  
INSERT INTO BOOK_COPIES VALUES (10, 1, 10), (5, 1, 11), (2, 2, 12), (5, 2, 13), (7, 3, 14), (1, 5, 10), (3, 4, 11);  
INSERT INTO CARD VALUES (100), (101),  
(102), (103), (104);  
INSERT INTO BOOK_LENDING VALUES ('2017-01-01', '2017-06-01', 1, 10, 101), ('2017-01-11', '2017-03-11',  
3, 14, 101),  
(('2017-02-21', '2017-04-21', 2, 13, 101), ('2017-03-15', '2017-07-15', 4, 11, 101), ('2017-04-12', '2017-05-  
12', 1, 11, 104));  
  
SELECT * FROM PUBLISHER;
```

	NAME	PHONE	ADDRESS
▶	GRUPO PLANETA	7756120238	BANGALORE
	HACHETTE LIVRE	8970862340	CHENAI
	MCGRAW-HILL	9989076587	BANGALORE
	PEARSON	9889076565	NEWDELHI
	RANDOM HOUSE	7455679345	HYDRABAD
✳	NULL	NULL	NULL

SELECT * FROM BOOK;

	BOOK_ID	TITLE	PUB_YEAR	PUBLISHER_NAME
▶	1	DBMS	JAN-2017	MCGRAW-HILL
	2	ADBMS	JUN-2016	MCGRAW-HILL
	3	CN	SEP-2016	PEARSON
	4	CG	SEP-2015	GRUPO PLANETA
	5	OS	MAY-2016	PEARSON
✳	NULL	NULL	NULL	NULL

SELECT * FROM BOOK_AUTHORS;

	AUTHOR_NAME	BOOK_ID
▶	NAVATHE	1
	NAVATHE	2
	TANENBAUM	3
	EDWARD ANGEL	4
	GALVIN	5
✳	NULL	NULL

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

SELECT * FROM CARD;

	CARD_NO
▶	100
	101
	102
	103
	104
*	NULL

SELECT * FROM BOOK_LENDING;

	DATE_OUT	DUE_DATE	BOOK_ID	BRANCH_ID	CARD_NO
▶	2017-01-01	2017-06-01	1	10	101
	2017-04-12	2017-05-12	1	11	104
	2017-02-21	2017-04-21	2	13	101
	2017-01-11	2017-03-11	3	14	101
	2017-03-15	2017-07-15	4	11	101
*	NULL	NULL	NULL	NULL	NULL

```
SELECT B.BOOK_ID, B.TITLE, B.PUBLISHER_NAME, A.AUTHOR_NAME,
C.NO_OF_COPIES, L.BRANCH_ID
FROM BOOK B, BOOK_AUTHORS A, BOOK_COPIES C, LIBRARY_BRANCH L
```

```

WHERE B.BOOK_ID=A.BOOK_ID
AND B.BOOK_ID=C.BOOK_ID
AND L.BRANCH_ID=C.BRANCH_ID;

```

	BOOK_ID	TITLE	PUBLISHER_NAME	AUTHOR_NAME	NO_OF_COPIES	BRANCH_ID
▶	1	DBMS	MCGRAW-HILL	NAVATHE	10	10
	1	DBMS	MCGRAW-HILL	NAVATHE	5	11
	2	ADBMS	MCGRAW-HILL	NAVATHE	2	12
	2	ADBMS	MCGRAW-HILL	NAVATHE	5	13
	3	CN	PEARSON	TANENBAUM	7	14
	4	CG	GRUPO PLANETA	EDWARD ANGEL	3	11
	5	OS	PEARSON	GALVIN	1	10

```

SELECT CARD_NO
FROM BOOK_LENDING
WHERE DATE_OUT BETWEEN '2017-01-01' AND '2017-07-01'
GROUP BY CARD_NO HAVING COUNT(*)>3;

```

	CARD_NO
▶	101

```

DELETE FROM BOOK
WHERE BOOK_ID=3;

```

```

CREATE VIEW V_PUBLICATION AS
SELECT PUB_YEAR
FROM BOOK;

```

```

SELECT * FROM V_PUBLICATION;

```

	PUB_YEAR
▶	JAN-2017
	JUN-2016
	SEP-2016
	SEP-2015
	MAY-2016

```
CREATE VIEW V_BOOKS AS
SELECT B.BOOK_ID, B.TITLE, C.NO_OF_COPIES
FROM BOOK B, BOOK_COPIES C, LIBRARY_BRANCH L
WHERE B.BOOK_ID=C.BOOK_ID
AND C.BRANCH_ID=L.BRANCH_ID;
```

```
SELECT * FROM V_BOOKS;
```

	BOOK_ID	TITLE	NO_OF_COPIES
▶	1	DBMS	10
	1	DBMS	5
	2	ADBMS	2
	2	ADBMS	5
	3	CN	7
	4	CG	3
	5	OS	1

Program 8:

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.

```
INSERT INTO student (regno, name, major, bdate) VALUES
    ('1pe11cs002', 'b', 'sr', '19930924'),
    ('1pe11cs003', 'c', 'sr', '19931127'),
    ('1pe11cs004', 'd', 'sr', '19930413'),
    ('1pe11cs005', 'e', 'jr', '19940824');
```

```
INSERT INTO course VALUES (111, 'OS', 'CSE'),
    (112, 'EC', 'CSE'),
    (113, 'SS', 'ISE'),
    (114, 'DBMS', 'CSE'),
    (115, 'SIGNALS', 'ECE');
```

```
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');
```

```
INSERT INTO enroll (regno, courseno, sem, marks) VALUES
    ('1pe11cs001', 115, 3, 100),
    ('1pe11cs002', 114, 5, 100),
    ('1pe11cs003', 113, 5, 100),
    ('1pe11cs004', 111, 5, 100),
    ('1pe11cs005', 112, 3, 100);
```

```
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);
```

Program :

```
CREATE DATABASE books3;  
USE books3;  
CREATE TABLE student(  
regno VARCHAR(15),  
name VARCHAR(20),  
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(15),  
courseno INT,  
sem INT,  
marks INT,  
PRIMARY KEY (regno,courseno),
```

```
FOREIGN KEY (regno) REFERENCES student(regno)
on update cascade on delete cascade,
FOREIGN KEY (courseno) REFERENCES course(courseno)
on update cascade on delete cascade);
CREATE TABLE text(
book_isbn INT,
book_title VARCHAR(20),
publisher VARCHAR(20),
author VARCHAR(20),
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)
on update cascade on delete cascade,
FOREIGN KEY (book_isbn) REFERENCES text(book_isbn)
on update cascade on delete cascade );
INSERT INTO student (regno,name,major,bdate) VALUES
('1pe11cs001','a','sr','19930926'),
('1pe11cs002','b','sr','19930924'),
('1pe11cs003','c','sr','19931127'),
('1pe11cs004','d','sr','19930413'),
('1pe11cs005','e','jr','19940824');
select * from student;
```

	regno	name	major	bdate
▶	1pe11cs001	a	sr	1993-09-26
	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
(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
▶	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	OS1	TATA MAC	HOBERT
*	NULL	NULL	NULL	NULL

```
INSERT INTO enroll (regno,courseno,sem,marks) VALUES  
('1pe11cs001',115,3,100),  
('1pe11cs002',114,5,100),  
('1pe11cs003',113,5,100),  
('1pe11cs004',111,5,100),  
('1pe11cs005',112,3,100);
```

```
select * from enroll;
```

	regno	courseno	sem	marks
▶	1pe11cs001	115	3	100
	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

```
Insert into text values (907,'OS1','TATA MAC','HOBERT');
```

```
SELECT c.courseno,t.book_isbn,t.book_title  
FROM course c,book_adoption ba,text t  
WHERE c.courseno=ba.courseno  
AND ba.book_isbn=t.book_isbn  
AND c.dept='CSE'  
AND 2<(  
SELECT COUNT(book_isbn)  
FROM book_adoption b  
WHERE c.courseno=b.courseno)  
ORDER BY t.book_title;
```

	courseno	book_isbn	book_title
▶	111	904	DATABASE SYSTEMS
	111	900	OPERATING SYS
	111	903	SCHEDULING

```
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='PEARSON')  
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 != 'PEARSON');
```

	dept

Program 9: Movie database

Consider the schema for Movie Database:

ACTOR (*Act_id*, *Act_Name*, *Act_Gender*)

DIRECTOR (*Dir_id*, *Dir_Name*, *Dir_Phone*)

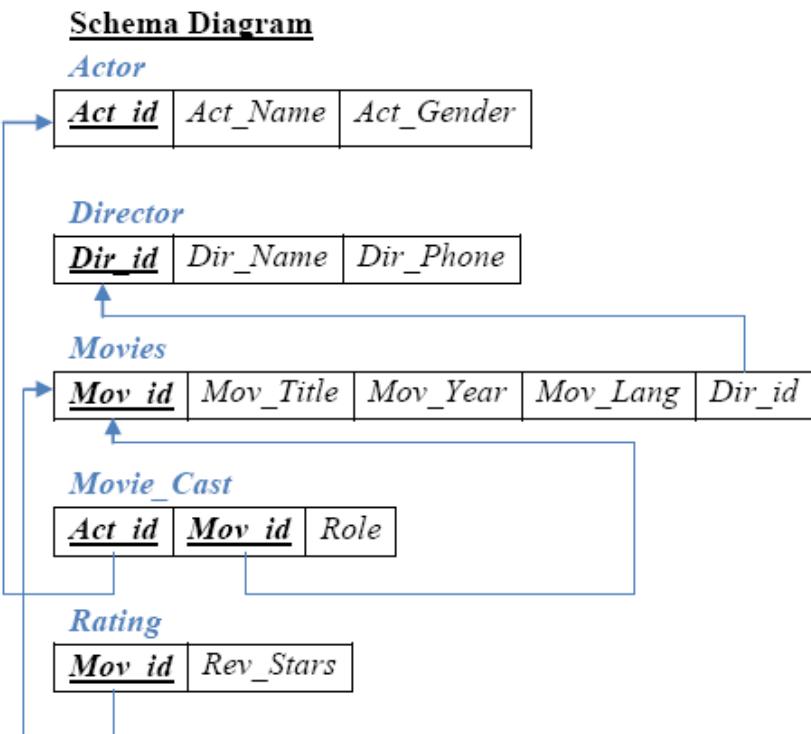
MOVIES (*Mov_id*, *Mov_Title*, *Mov_Year*, *Mov_Lang*, *Dir_id*)

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.



```
SQL> SELECT * FROM ACTOR;
```

ACT_ID	ACT_NAME	A
301	ANUSHKA	F
302	PRABHAS	M
303	PUNITH	M
304	JERMY	M

```
SQL> SELECT * FROM DIRECTOR;
```

DIR_ID	DIR_NAME	DIR_PHONE
60	RAJAMOULI	8751611001
61	HITCHCOCK	7766138911
62	FARAN	9986776531
63	STEVEN SPIELBERG	8989776530

```
SQL> SELECT * FROM MOVIES;
```

MOV_ID	MOV_TITLE	MOV_YEAR	MOV_LANG	DIR_ID
1001	BAHUBALI-2	2017	TELAGU	60
1002	BAHUBALI-1	2015	TELAGU	60
1003	AKASH	2008	KANNADA	61
1004	WAR HORSE	2011	ENGLISH	63

```
SQL> SELECT * FROM MOVIE_CAST;
```

ACT_ID	MOV_ID	ROLE
301	1002	HEROINE
301	1001	HEROINE
303	1003	HERO
303	1002	GUEST
304	1004	HERO

```
SQL> SELECT * FROM RATING;
```

MOV_ID	REV_STARS
1001	4
1002	2
1003	5
1004	4

Program :

```
create database movie;
use movie;

CREATE TABLE ACTOR (
    ACT_ID INT,
    ACT_NAME VARCHAR (20),
    ACT_GENDER CHAR(1),
    PRIMARY KEY (ACT_ID));

CREATE TABLE DIRECTOR (
    DIR_ID INT,
    DIR_NAME VARCHAR (20),
    DIR_PHONE VARCHAR(10),
    PRIMARY KEY (DIR_ID));

CREATE TABLE MOVIES (
    MOV_ID INT,
    MOV_TITLE VARCHAR (25),
    MOV_YEAR INT,
    MOV_LANG VARCHAR (12),
    DIR_ID INT,
    PRIMARY KEY (MOV_ID),
    FOREIGN KEY (DIR_ID) REFERENCES DIRECTOR (DIR_ID)
    ON UPDATE CASCADE ON DELETE CASCADE);

CREATE TABLE MOVIE_CAST (
    ACT_ID INT,
    MOV_ID INT,
    ROLE VARCHAR(10),
    PRIMARY KEY (ACT_ID, MOV_ID),
    FOREIGN KEY (ACT_ID) REFERENCES ACTOR (ACT_ID)
```

```

ON UPDATE CASCADE ON DELETE CASCADE,
FOREIGN KEY (MOV_ID) REFERENCES MOVIES (MOV_ID)
ON UPDATE CASCADE ON DELETE CASCADE);
CREATE TABLE RATING (
MOV_ID INT,
REV_STARS VARCHAR(25),
PRIMARY KEY (MOV_ID),
FOREIGN KEY (MOV_ID) REFERENCES MOVIES (MOV_ID)
ON UPDATE CASCADE ON DELETE CASCADE);

```

```

INSERT INTO ACTOR VALUES (301,'ANUSHKA','F');
INSERT INTO ACTOR VALUES (302,'PRABHAS','M');
INSERT INTO ACTOR VALUES (303,'PUNITH','M');
INSERT INTO ACTOR VALUES (304,'JERMY','M');

```

```
SELECT * FROM ACTOR;
```

	ACT_ID	ACT_NAME	ACT_GENDER
▶	301	ANUSHKA	F
	302	PRABHAS	M
	303	PUNITH	M
*	304	JERMY	M
	NULL	NULL	NULL

```

INSERT INTO DIRECTOR VALUES (60,'RAJAMOULI', '8751611001');
INSERT INTO DIRECTOR VALUES (61,'HITCHCOCK', '7766138911');
INSERT INTO DIRECTOR VALUES (62,'FARAN', '9986776531');
INSERT INTO DIRECTOR VALUES (63,'STEVEN SPIELBERG', '8989776530');

```

```
SELECT * FROM DIRECTOR;
```

	DIR_ID	DIR_NAME	DIR_PHONE
▶	60	RAJAMOULI	8751611001
	61	HITCHCOCK	7766138911
	62	FARAN	9986776531
*	63	STEVEN SPIELBERG	8989776530
	NULL	NULL	NULL

```
INSERT INTO MOVIES VALUES (1001,'BAHUBALI-2', 2017, 'TELAGU', 60);
```

```
INSERT INTO MOVIES VALUES (1002,'AHUBALI-1', 2015, 'TELAGU', 60);
```

```
INSERT INTO MOVIES VALUES (1003,'AKASH', 2008, 'KANNADA', 61);
```

```
INSERT INTO MOVIES VALUES (1004,'WAR HORSE', 2011, 'ENGLISH', 63);
```

```
SELECT * FROM MOVIES;
```

	MOV_ID	MOV_TITLE	MOV_YEAR	MOV_LANG	DIR_ID
▶	1001	BAHUBALI-2	2017	TELAGU	60
	1002	AHUBALI-1	2015	TELAGU	60
	1003	AKASH	2008	KANNADA	61
*	1004	WAR HORSE	2011	ENGLISH	63
	NULL	NULL	NULL	NULL	NULL

```
INSERT INTO MOVIE_CAST VALUES (301, 1002, 'HEROINE');
```

```
INSERT INTO MOVIE_CAST VALUES (301, 1001, 'HEROINE');
```

```
INSERT INTO MOVIE_CAST VALUES (303, 1003, 'HERO');
```

```
INSERT INTO MOVIE_CAST VALUES (303, 1002, 'GUEST');
```

```
INSERT INTO MOVIE_CAST VALUES (304, 1004, 'HERO');
```

```
SELECT * FROM MOVIE_CAST;
```

	ACT_ID	MOV_ID	ROLE
▶	301	1001	HEROINE
	301	1002	HEROINE
	303	1002	GUEST
	303	1003	HERO
*	304	1004	HERO
	NULL	NULL	NULL

```

INSERT INTO RATING VALUES (1001, 4);
INSERT INTO RATING VALUES (1002, 2);
INSERT INTO RATING VALUES (1003, 5);
INSERT INTO RATING VALUES (1004, 4);
SELECT * FROM RATING;

```

	MOV_ID	REV_STARS
▶	1001	4
	1002	2
	1003	5
	1004	4
*	NULL	NULL

```

SELECT MOV_TITLE
FROM MOVIES
WHERE DIR_ID IN (SELECT DIR_ID
FROM DIRECTOR
WHERE DIR_NAME = 'HITCHCOCK');

```

	MOV_TITLE
▶	AKASH

```

SELECT MOV_TITLE
FROM MOVIES M, MOVIE_CAST MV
WHERE M.MOV_ID=MV.MOV_ID AND ACT_ID IN (SELECT ACT_ID
FROM MOVIE_CAST GROUP BY ACT_ID
HAVING COUNT(ACT_ID)>1)
GROUP BY MOV_TITLE
HAVING COUNT(*)>1;

```

	MOV_TITLE
▶	AHUBALI-1

```

SELECT ACT_NAME, MOV_TITLE, MOV_YEAR
FROM ACTOR A
JOIN MOVIE_CAST C
ON A.ACT_ID=C.ACT_ID
JOIN MOVIES M
ON C.MOV_ID=M.MOV_ID
WHERE M.MOV_YEAR NOT BETWEEN 2000 AND 2015;

```

	ACT_NAME	MOV_TITLE	MOV_YEAR
▶	ANUSHKA	BAHUBALI-2	2017

```

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)
▶	AHUBALI-1	2
	AKASH	5
	BAHUBALI-2	4
	WAR HORSE	4

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
▶	1001	4
	1002	2
	1003	5
	1004	5
*	NULL	NULL

Program 10

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:

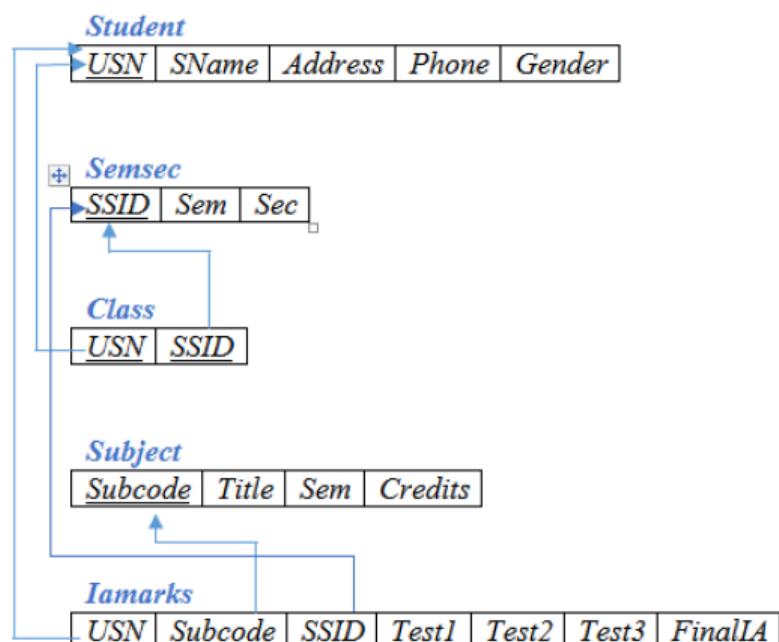
If FinalIA = 17 to 20 then CAT = ‘Outstanding’

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



```
SQL> SELECT * FROM IAMARKS;
```

USN	SUBCODE	SSID	TEST1	TEST2	TEST3	FINALIA
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	

SQL> SELECT * FROM STUDENT1;

USN	SNAME	ADDRESS	PHONE	G
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
1RN15CS011	AJAY	TUMKUR	9845091341	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	CHIKAMAGALUR	8800880011	M
1RN14CS025	ASMI	BENGALURU	7894737377	F

SQL> SELECT * FROM SEMSEC;

SSID	SEM	S
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
CSE2C	2	C
CSE2B	2	B
CSE1A	1	A
CSE1B	1	B
CSE1C	1	C

SQL> SELECT * FROM CLASS;

USN	SSID
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

14 rows selected.

SUBCODE	TITLE	SEM	CREDITS
10CS81	ACA	8	4
10CS82	SSM	8	4
10CS83	NM	8	4
10CS84	CC	8	4
10CS85	PW	8	4
10CS71	OODAD	7	4
10CS72	ECS	7	4
10CS73	PTW	7	4
10CS74	DWDM	7	4
10CS75	JAVA	7	4
10CS76	SAN	7	4
15CS51	ME	5	4
15CS52	CN	5	4
15CS53	DBMS	5	4
15CS54	ATC	5	4
15CS55	JAVA	5	3
15CS56	AI	5	3
15CS41	M4	4	4
15CS42	SE	4	4
15CS43	DAA	4	4
15CS44	MPMC	4	4
15CS45	OOC	4	3
15CS46	DC	4	3
15CS31	M3	3	4
15CS32	ADE	3	4
15CS33	DSA	3	4
15CS34	CO	3	4
15CS35	USP	3	3
15CS36	DMS	3	3

Program :

```
CREATE DATABASE COLLEGE;  
USE COLLEGE;  
CREATE TABLE STUDENT (  
    USN VARCHAR (10),  
    SNAME VARCHAR (25),  
    ADDRESS VARCHAR (25),  
    PHONE VARCHAR(10),  
    GENDER CHAR (1),  
    PRIMARY KEY(USN));  
CREATE TABLE SEMSEC (  
    SSID VARCHAR (5),  
    SEM INT,  
    SEC CHAR (1),  
    PRIMARY KEY(SSID));  
CREATE TABLE CLASS (  
    USN VARCHAR (10),  
    SSID VARCHAR (5),  
    PRIMARY KEY (USN, SSID),  
    FOREIGN KEY (USN) REFERENCES STUDENT (USN)  
    ON UPDATE CASCADE ON DELETE CASCADE,  
    FOREIGN KEY (SSID) REFERENCES SEMSEC (SSID)  
    ON UPDATE CASCADE ON DELETE CASCADE);  
CREATE TABLE SUBJECT (  
    SUBCODE VARCHAR (8),  
    TITLE VARCHAR (20),  
    SEM INT,
```

```
CREDITS INT,  
PRIMARY KEY (SUBCODE));  
  
CREATE TABLE IAMARKS (  
USN VARCHAR (10),  
SUBCODE VARCHAR (8),  
SSID VARCHAR (5),  
TEST1 INT,  
TEST2 INT,  
TEST3 INT,  
FINALIA INT,  
PRIMARY KEY (USN, SUBCODE, SSID),  
FOREIGN KEY (USN) REFERENCES STUDENT (USN)  
ON UPDATE CASCADE ON DELETE CASCADE,  
FOREIGN KEY (SUBCODE) REFERENCES SUBJECT (SUBCODE)  
ON UPDATE CASCADE ON DELETE CASCADE,  
FOREIGN KEY (SSID) REFERENCES SEMSEC (SSID)  
ON UPDATE CASCADE ON DELETE CASCADE);  
  
INSERT INTO STUDENT VALUES ('1RN13CS020','AKSHAY','BELAGAVI', 8877881122,'M');  
INSERT INTO STUDENT VALUES ('1RN13CS062','SANDHYA','BENGALURU',  
7722829912,'F');  
INSERT INTO STUDENT VALUES ('1RN13CS091','TEESHA','BENGALURU', 7712312312,'F');  
INSERT INTO STUDENT VALUES ('1RN13CS066','SUPRIYA','MANGALURU', 8877881122,'F');  
INSERT INTO STUDENT VALUES ('1RN14CS010','ABHAY','BENGALURU', 9900211201,'M');  
INSERT INTO STUDENT VALUES ('1RN14CS032','BHASKAR','BENGALURU','9923211099','M');  
INSERT INTO STUDENT VALUES ('1RN14CS025','ASMI','BENGALURU', '7894737377','F');  
INSERT INTO STUDENT VALUES ('1RN15CS011','AJAY','TUMKUR', '9845091341','M');  
INSERT INTO STUDENT VALUES ('1RN15CS029','CHITRA','DAVANGERE', '7696772121','F');  
INSERT INTO STUDENT VALUES ('1RN15CS045','JEEVA','BELLARY', '9944850121','M');
```

```

INSERT INTO STUDENT VALUES ('1RN15CS091','SANTOSH','MANGALURU',
'8812332201','M');

INSERT INTO STUDENT VALUES ('1RN16CS045','ISMAIL','KALBURGI', '9900232201','M');

INSERT INTO STUDENT VALUES ('1RN16CS088','SAMEERA','SHIMOGA', '9905542212','F');

INSERT INTO STUDENT VALUES ('1RN16CS122','VINAYAKA','CHIKAMAGALUR',
'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	9845091341	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
●	HULL	HULL	HULL	HULL	HULL

```

INSERT INTO SEMSEC VALUES ('CSE8A', 8,'A');

INSERT INTO SEMSEC VALUES ('CSE8B', 8,'B');

INSERT INTO SEMSEC VALUES ('CSE8C', 8,'C');

INSERT INTO SEMSEC VALUES ('CSE7A', 7,'A');

INSERT INTO SEMSEC VALUES ('CSE7B', 7,'B');

INSERT INTO SEMSEC VALUES ('CSE7C', 7,'C');

INSERT INTO SEMSEC VALUES ('CSE6A', 6,'A');

INSERT INTO SEMSEC VALUES ('CSE6B', 6,'B');

```

```
INSERT INTO SEMSEC VALUES ('CSE6C', 6,'C');
INSERT INTO SEMSEC VALUES ('CSE5A', 5,'A');
INSERT INTO SEMSEC VALUES ('CSE5B', 5,'B');
INSERT INTO SEMSEC VALUES ('CSE5C', 5,'C');
INSERT INTO SEMSEC VALUES ('CSE4A', 4,'A');
INSERT INTO SEMSEC VALUES ('CSE4B', 4,'B');
INSERT INTO SEMSEC VALUES ('CSE4C', 4,'C');
INSERT INTO SEMSEC VALUES ('CSE3A', 3,'A');
INSERT INTO SEMSEC VALUES ('CSE3B', 3,'B');
INSERT INTO SEMSEC VALUES ('CSE3C', 3,'C');
INSERT INTO SEMSEC VALUES ('CSE2A', 2,'A');
INSERT INTO SEMSEC VALUES ('CSE2B', 2,'B');
INSERT INTO SEMSEC VALUES ('CSE2C', 2,'C');
INSERT INTO SEMSEC VALUES ('CSE1A', 1,'A');
INSERT INTO SEMSEC VALUES ('CSE1B', 1,'B');
INSERT INTO SEMSEC VALUES ('CSE1C', 1,'C');
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');

INSERT INTO CLASS VALUES ('1RN13CS062','CSE8A');

INSERT INTO CLASS VALUES ('1RN13CS066','CSE8B');

INSERT INTO CLASS VALUES ('1RN13CS091','CSE8C');

INSERT INTO CLASS VALUES ('1RN14CS010','CSE7A');

INSERT INTO CLASS VALUES ('1RN14CS025','CSE7A');

INSERT INTO CLASS VALUES ('1RN14CS032','CSE7A');

```

```

INSERT INTO CLASS VALUES ('1RN15CS011','CSE4A');

INSERT INTO CLASS VALUES ('1RN15CS029','CSE4A');

INSERT INTO CLASS VALUES ('1RN15CS045','CSE4B');

INSERT INTO CLASS VALUES ('1RN15CS091','CSE4C');

INSERT INTO CLASS VALUES ('1RN16CS045','CSE3A');

INSERT INTO CLASS VALUES ('1RN16CS088','CSE3B');

INSERT INTO CLASS VALUES ('1RN16CS122','CSE3C');

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);

INSERT INTO SUBJECT VALUES ('10CS82','SSM', 8, 4);

INSERT INTO SUBJECT VALUES ('10CS83','NM', 8, 4);

INSERT INTO SUBJECT VALUES ('10CS84','CC', 8, 4);

INSERT INTO SUBJECT VALUES ('10CS85','PW', 8, 4);

INSERT INTO SUBJECT VALUES ('10CS71','OOAD', 7, 4);

INSERT INTO SUBJECT VALUES ('10CS72','ECS', 7, 4);

```

```
INSERT INTO SUBJECT VALUES ('10CS73','PTW', 7, 4);
INSERT INTO SUBJECT VALUES ('10CS74','DWDM', 7, 4);
INSERT INTO SUBJECT VALUES ('10CS75','JAVA', 7, 4);
INSERT INTO SUBJECT VALUES ('10CS76','SAN', 7, 4);
INSERT INTO SUBJECT VALUES ('15CS51', 'ME', 5, 4);
INSERT INTO SUBJECT VALUES ('15CS52','CN', 5, 4);
INSERT INTO SUBJECT VALUES ('15CS53','DBMS', 5, 4);
INSERT INTO SUBJECT VALUES ('15CS54','ATC', 5, 4);
INSERT INTO SUBJECT VALUES ('15CS55','JAVA', 5, 3);
INSERT INTO SUBJECT VALUES ('15CS56','AI', 5, 3);
INSERT INTO SUBJECT VALUES ('15CS41','M4', 4, 4);
INSERT INTO SUBJECT VALUES ('15CS42','SE', 4, 4);
INSERT INTO SUBJECT VALUES ('15CS43','DAA', 4, 4);
INSERT INTO SUBJECT VALUES ('15CS44','MPMC', 4, 4);
INSERT INTO SUBJECT VALUES ('15CS45','OOC', 4, 3);
INSERT INTO SUBJECT VALUES ('15CS46','DC', 4, 3);
INSERT INTO SUBJECT VALUES ('15CS31','M3', 3, 4);
INSERT INTO SUBJECT VALUES ('15CS32','ADE', 3, 4);
INSERT INTO SUBJECT VALUES ('15CS33','DSA', 3, 4);
INSERT INTO SUBJECT VALUES ('15CS34','CO', 3, 4);
INSERT INTO SUBJECT VALUES ('15CS35','USP', 3, 3);
INSERT INTO SUBJECT VALUES ('15CS36','DMS', 3, 3);
SELECT * FROM SUBJECT;
```

	SUBCODE	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
	15CS31	M3	3	4
	15CS32	ADE	3	4
	15CS33	DSA	3	4
	15CS34	CO	3	4
	15CS35	USP	3	3
	15CS36	DMS	3	3
	15CS41	M4	4	4
	15CS42	SE	4	4
	15CS43	DAA	4	4
	15CS44	MPMC	4	4
	15CS45	OOC	4	3
	15CS46	DC	4	3
	15CS51	ME	5	4
	15CS52	CN	5	4
	15CS53	DBMS	5	4
	15CS54	ATC	5	4
	15CS55	JAVA	5	3
	15CS56	AI	5	3
*	HULL	HULL	HULL	HULL

INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES

('1RN13CS091','10CS81','CSE8C', 15, 16, 18);

INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES

('1RN13CS091','10CS82','CSE8C', 12, 19, 14);

```

INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES
('1RN13CS091','10CS83','CSE8C', 19, 15, 20);

INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES
('1RN13CS091','10CS84','CSE8C', 20, 16, 19);

INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES
('1RN13CS091','10CS85','CSE8C', 15, 15, 12);

SELECT * FROM IAMARKS;

```

	USN	SUBCODE	SSID	TEST1	TEST2	TEST3	FINALIA
▶	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

```

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

```

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

```

CREATE VIEW STU_TEST1_MARKS_VIEW
AS
SELECT TEST1, SUBCODE
FROM IAMARKS
WHERE USN = '1RN13CS091';

SELECT * FROM STU_TEST1_MARKS_VIEW;

```

	TEST1	SUBCODE
▶	15	10CS81
	12	10CS82
	19	10CS83
	20	10CS84
	15	10CS85

```

SELECT S.USN,S.SNAME,S.ADDRESS,S.PHONE,S.GENDER,
(CASE
WHEN IA.FINALIA BETWEEN 17 AND 20 THEN 'OUTSTANDING'
WHEN IA.FINALIA BETWEEN 12 AND 16 THEN 'AVERAGE'
ELSE 'WEAK'
END) AS CAT
FROM STUDENT S, SEMSEC SS, IAMARKS IA, SUBJECT SUB
WHERE S.USN = IA.USN AND
SS.SSID = IA.SSID AND
SUB.SUBCODE = IA.SUBCODE AND
SUBSEM = 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