Consider the following database for a banking enterprise.  
BRANCH (branch-name: String, branch-city: String, assets: real)  
ACCOUNTS (accno: int, branch-name: String, balance: real)  
DEPOSITOR (customer-name: String, accno:int)  
CUSTOMER( customer-name: String,customer-street: String, customer-city: String) LOAN (loan-number: int, branch-name: String, amount: real)  
BORROWER (customer-name: String, loan-number: int)

create database Banking\_enterprise;

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

create table branch(  
branch\_name varchar(30) primary key, branch\_city varchar(30),

assets real);

create table accounts(  
accno int primary key, branch\_name varchar(30), balance real,

foreign key (branch\_name) references branch(branch\_name) on delete cascade on update cascade);

create table customer(  
customer\_name varchar(30) primary key, customer\_street varchar(20), customer\_city varchar(20));

create table depositor(  
customer\_name varchar(30),  
accno int,  
primary key(customer\_name ,accno),  
foreign key (accno) references accounts(accno) on delete cascade on update cascade,  
foreign key (customer\_name) references customer(customer\_name) on delete cascade on update

cascade);

create table loan(  
loan\_number int primary key,  
branch\_name varchar(30),  
amount real,  
foreign key (branch\_name) references branch(branch\_name)

);

create table borrower (  
customer\_name varchar(30),  
loan\_number int,  
primary key(customer\_name, loan\_number),  
foreign key (customer\_name) references customer(customer\_name) on delete cascade on update cascade, foreign key (loan\_number) references loan(loan\_number) on delete cascade on update cascade);

ii. Enter at least five tuples for each relation.

insert into branch(branch\_name,branch\_city,assets) values ('A','Bangalore',190000),  
('B','Bangalore',200000),  
('C','Delhi',235344),

('D','Chennai',1050560), ('E','Chennai',678909);

insert into accounts(accno,branch\_name,balance) VALUES (1001,'A',10000),  
(1002,'B',5000),  
(1003,'C',7500),

(1004,'D',50000),

(1005,'D',75000), (1006,'E',560); (1007,"B",500), (1008,"B",1500);

insert into customer(customer\_name,customer\_street,customer\_city) VALUES ("Ravi","Dasarahalli","Bangalore"),  
("Shyam","Indiranagar","Delhi"),  
("Seema","Vasantnagar","Chennai"),

("Arpita","Church Street","Bangalore"), ("Vinay","MG Road","Chennai");

insert into depositor(customer\_name,accno) VALUES ("Ravi",1001),  
("Ravi",1002),  
("Shyam",1003),

("Seema",1004), ("Seema",1005), ("Arpita",1006), ("Vinay",1007), ("Vinay",1008);

insert into loan(loan\_number,branch\_name,amount) VALUES (001,'A',10000),  
(002,'B',25000),  
(003,'B',250000),

(004,'C',5000), (005,'E',90000),

insert into borrower(customer\_name,loan\_number) VALUES ("Arpita",001),  
("Ravi",002),  
("Arpita",003),

("Shyam",004), ("Vinay",005);

iii. Find all the customers who have at least two accounts at the Main branch.

select customer\_name from depositor  
join accounts on depositor.accno = accounts.accno where accounts.branch\_name = "D" group by depositor.customer\_name having count(depositor.customer\_name) >=2;

+---------------+

| customer\_name |

+---------------+

| Seema |

+---------------+

1 row in set (0.01 sec)

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

select customer\_name from depositor  
join accounts on accounts.accno = depositor.accno  
join branch on branch.branch\_name = accounts.branch\_name  
where branch.branch\_city = "Bangalore"  
GROUP BY depositor.customer\_name  
having count(DISTINCT branch.branch\_name) = (SELECT COUNT(branch\_name)

FROM branch  
WHERE branch\_city = 'Bangalore');

+---------------+

| customer\_name |

+---------------+

| Ravi |

+---------------+

1 row in set (0.00 sec)

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

delete from accounts where branch\_name in  
(select branch\_name from branch where branch\_city="Delhi");

Query OK, 1 row affected (0.02 sec)