## Bank Database

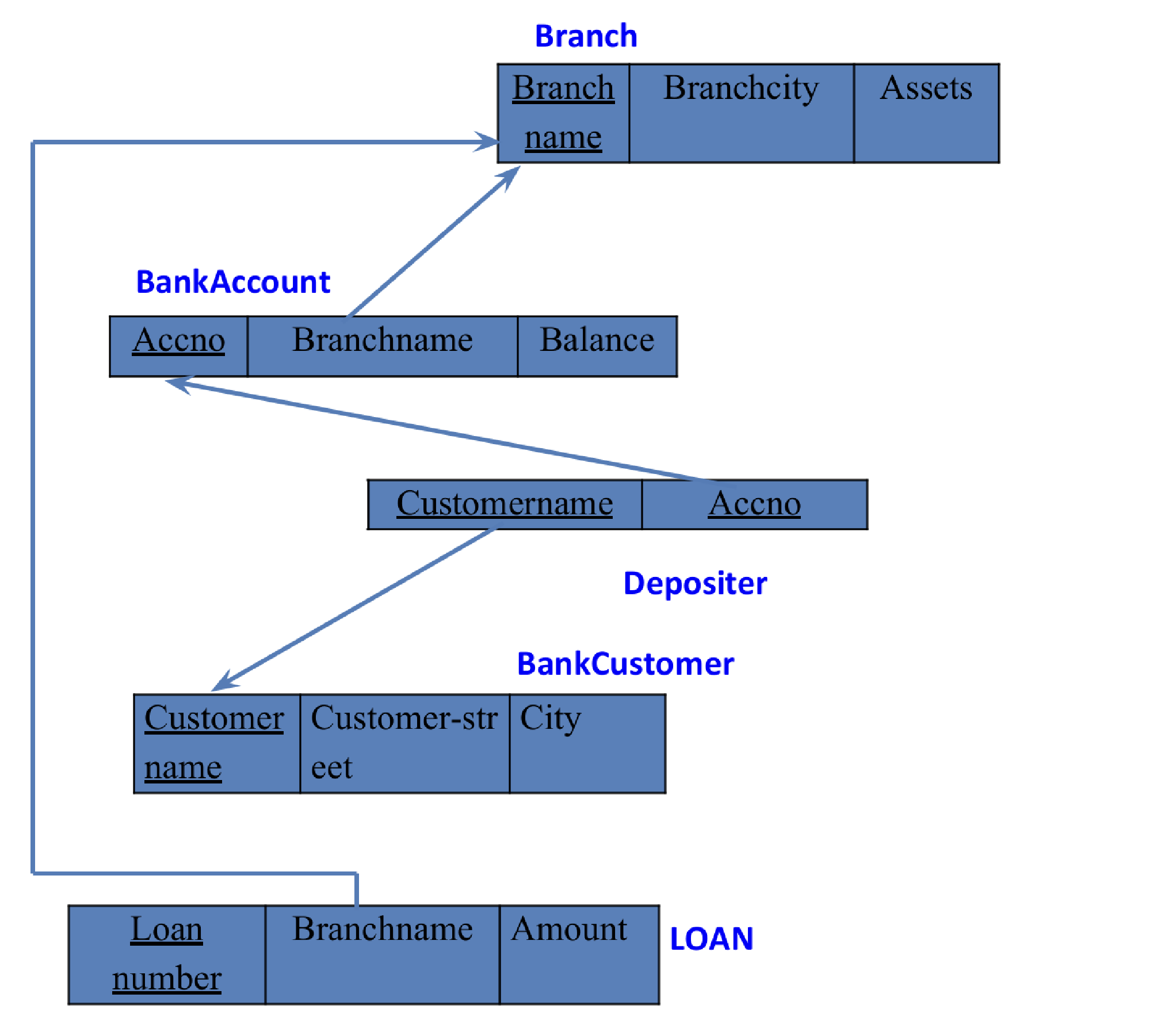
**Question**

* 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)
* Create the above tables by properly specifying the primary keys and the foreign keys. Enter at least five tuples for each relation.
* Display the branch name and assets from all branches in lakhs of rupees and rename the assets column to 'assets in lakhs'.
* Find all the customers who have at least two accounts at the same branch (ex. SBI\_ResidencyRoad).
* Create a viewwhich gives each branch the sum of the amount of all the loans at the branch.

### Schema Diagram



### Create database

### create database bank\_database\_1BM23CS081;

### use bank\_database\_1BM23CS081;

### Create table

### create table BRANCH\_081(

### branch\_name varchar(100),

### branch\_city varchar(100),

### assets real,

### primary key (branch\_name));

### create table BANK\_ACCOUNT\_081(

### acc\_no int,

### branch\_name varchar(100),

### balance real,

### primary key(acc\_no),

### foreign key(branch\_name) references BRANCH\_081(branch\_name));

### create table BANK\_CUSTOMER\_081(

### customer\_name varchar(100),

### customer\_street varchar(100),

### customer\_city varchar(100),

### primary key(customer\_name));

### create table DEPOSITOR\_081(

### customer\_name varchar(100),

### acc\_no int,

### primary key(customer\_name, acc\_no),

### foreign key(acc\_no) references BANK\_ACCOUNT\_081(acc\_no),

### foreign key(customer\_name) references BANK\_CUSTOMER\_081(customer\_name));

### create table LOAN\_081(

### loan\_number int,

### branch\_name varchar(100),

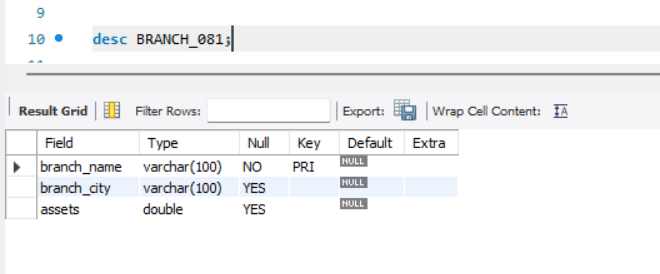
### amount real,

### primary key(loan\_number),

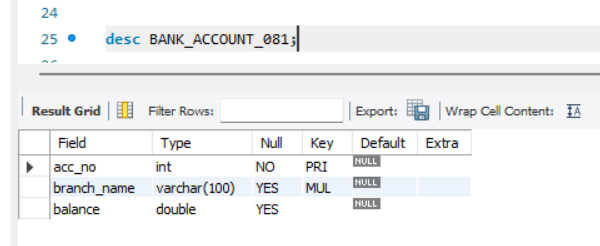
### foreign key(branch\_name) references BRANCH\_081(branch\_name));

### Structure of the table

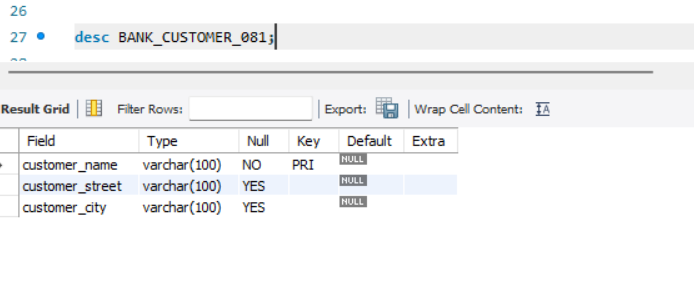
BRANCH\_081



BANK\_ACCOUNT\_081

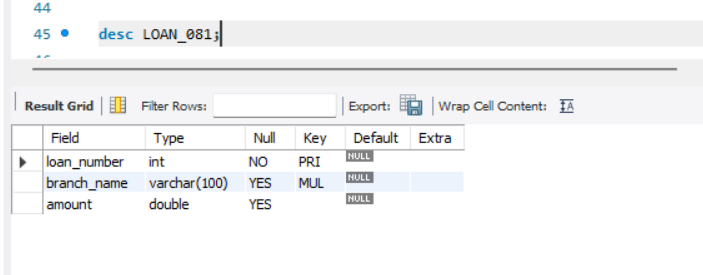


### BANK\_CUSTOMER\_081

DEPOSITOR\_081

### 

LOAN\_081



### Inserting Values to the table

insert into BRANCH\_081(branch\_name, branch\_city, assets)

values ("SBI\_Chamrajpet", "Bangalore", 50000),

("SBI\_ResidencyRoad", "Bangalore", 10000),

("SBI\_Shivaginagar", "Bombay", 20000),

("SBI\_ParlimentRoad", "Delhi", 10000),

("SBI\_Jantarmantar", "Delhi", 20000);

insert into BANK\_ACCOUNT\_081(acc\_no, branch\_name, balance)

values (1, "SBI\_Chamrajpet", 2000),

(2, "SBI\_ResidencyRoad", 5000),

(3, "SBI\_Shivaginagar", 6000),

(4, "SBI\_ParlimentRoad", 9000),

(5, "SBI\_Jantarmantar", 8000),

(6, "SBI\_Shivaginagar", 4000),

(8, "SBI\_ResidencyRoad", 4000),

(9, "SBI\_ParlimentRoad", 3000),

(10, "SBI\_ResidencyRoad", 5000),

(11, "SBI\_Jantarmantar", 2000);

insert into BANK\_CUSTOMER\_081(customer\_name, customer\_street,customer\_city)

values ("Avinash", "Bull Temple Road", "Bangalore"),

("Dinesh", "Bannergatta Road", "Bangalore"),

("Mohan", "National College Road", "Bangalore"),

("Nikil", "Akbar Road", "Delhi"),

("Ravi", "Prithiviraj Road", "Delhi");

insert into DEPOSITOR\_081(customer\_name, acc\_no)

values ("Avinash", 1),

("Dinesh", 2),

("Nikil", 4),

("Ravi", 5),

("Avinash", 8),

("Nikil", 9),

("Dinesh", 10),

("Nikil", 11);

insert into LOAN\_081(loan\_number, branch\_name, amount)

values (1, "SBI\_Chamrajpet", 1000),

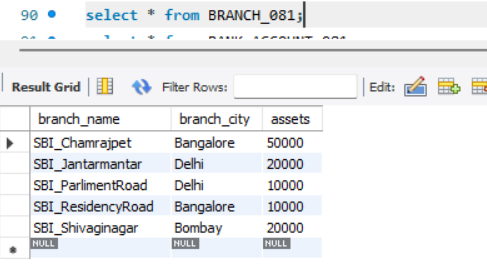
(2, "SBI\_ResidencyRoad", 2000),

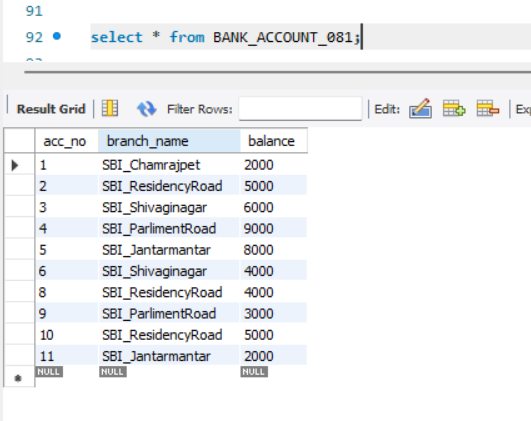
(3, "SBI\_Shivaginagar", 3000),

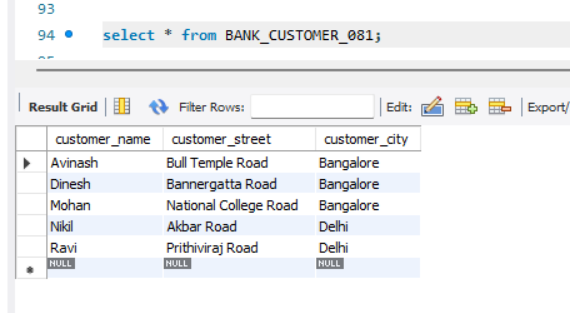
(4, "SBI\_ParlimentRoad", 4000),

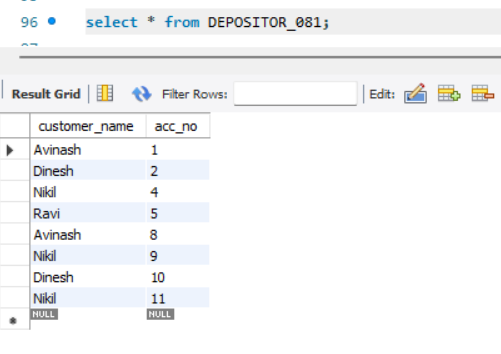
(5, "SBI\_Jantarmantar", 5000);

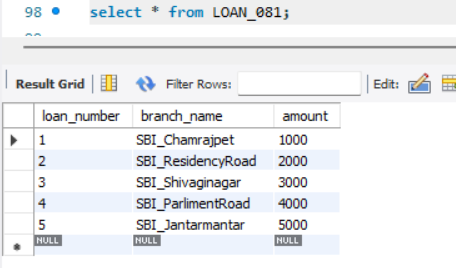
**Inserted Values**







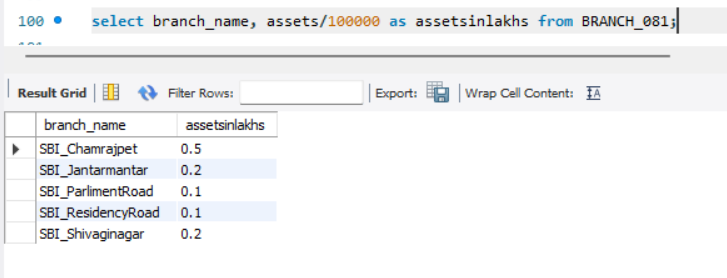




### Queries

**1.Display the branch name and assets from all branches and rename the assets column to 'assets in lakhs'.**

Query : select branch\_name, assets/100000 as assetsinlakhs from BRANCH\_081;



**2.Find all the customers who have at least two accounts at the same branch (ex.SBI\_ResidencyRoad).**

Query : select d.customer\_name, b.branch\_name, count(d.acc\_no) as account\_no

from DEPOSITOR\_081 d, BANK\_ACCOUNT\_081 b

where d.acc\_no=b.acc\_no

group by d.customer\_name, b.branch\_name

having count(d.acc\_no)>=2;

