Banking Example (primary key is underlined)

branch (branch-name, branch-city, assets)
customer (customer-name, customer-street, customer-city)

account (account-number, branch-name, balance)

loan (loan-number, branch-name, amount)

depositor (customer-name, account-number)

borrower (customer-name, loan-number)

employee (employee-name, branch-name, salary)

(1). Create database 'bank'

SQL: *create database if not exists bank*;

You can use command 'show database;' to check if 'bank' has been created.

(2). Create above tables in database 'bank'

SQL:
use bank;

```
create table if not exists account
(
account_number char(5) not null primary key,
branch_name varchar(10),
balance double
);

create table if not exists branch
(
branch_name varchar(10) not null primary key,
branch_city varchar(10),
assets double
);

create table if not exists customer
(
customer_name varchar(20) not null primary key,
customer_street varchar(20),
customer_city varchar(10)
);
```

```
create table if not exists loan
(
loan_number varchar(5) not null primary key,
branch_name varchar(10),
amount double
);

create table if not exists borrower
(
customer_name varchar(20) not null,
loan_number varchar(5) not null,
primary key(customer_name, loan_number)
);


create table if not exists depositor
(
customer_name varchar(20) not null,
account_number char(5) not null,
primary key(customer_name, account_number)
);


create table if not exists employee
(
employee_name varchar(20) not null,
branch_name varchar(10) not null,
salary double,
primary key(employee_name,branch_name)
);
```

Similarly, you can use command 'show tables;' to check if all tables have been created.

(3). **Insert data into those tables**

SQL:

```
use bank;

insert into account values('A-101', 'Downtown',   500);
insert into account values('A-102', 'Perryridge', 400);
insert into account values('A-201', 'Brighton',   900);
insert into account values('A-215', 'Mianus',     700);
insert into account values('A-217', 'Brighton',   750);
insert into account values('A-222', 'Redwood',    700);
insert into account values('A-305', 'Round Hill', 350);
```

```
insert into branch values('Brighton',   'Brooklyn',    7100000);
insert into branch values('Downtown',   'Brooklyn',    9000000);
insert into branch values('Mianus',     'Horseneck',   400000);
insert into branch values('North Town', 'Rye',         3700000);
insert into branch values('Perryridge', 'Horseneck',   1700000);
insert into branch values('Pownal',     'Bennington',  300000);
insert into branch values('Redwood',    'Palo Alto',   2100000);
insert into branch values('Round Hill', 'Horseneck',   8000000);


insert into customer values('Adams',    'Spring',     'Pittsfield');
insert into customer values('Brooks',   'Senator',    'Brooklyn');
insert into customer values('Curry',    'North',      'Rye');
insert into customer values('Glenn',    'Sand Hill',  'Woodside');
insert into customer values('Green',    'Walnut',     'Stamford');
insert into customer values('Hayes',    'Main',       'Harrison');
insert into customer values('Johnson',  'Alma',       'Palo Alto');
insert into customer values('Jones',    'Main',       'Harrison');
insert into customer values('Lindsay',  'Park',       'Pittsfield');
insert into customer values('Smith',    'North',      'Rye');
insert into customer values('Turner',   'Putnam',     'Stamford');
insert into customer values('Williams', 'Nassau',     'Princeton');



insert into depositor values('Hayes',    'A-102');
insert into depositor values('Johnson',  'A-102');
insert into depositor values('Johnson',  'A-201');
insert into depositor values('Jones',    'A-217');
insert into depositor values('Lindsay',  'A-222');
insert into depositor values('Smith',    'A-215');
insert into depositor values('Turner',   'A-305');


insert into loan values('L-11', 'Round Hill',    900);
insert into loan values('L-14', 'Downtown',     1500);
insert into loan values('L-15', 'Perryridge',   1500);
insert into loan values('L-16', 'Perryridge',   1300);
insert into loan values('L-17', 'Downtown',     1000);
insert into loan values('L-23', 'Redwood',      2000);
insert into loan values('l-93', 'Mianus',        500);

insert into borrower values('Adams',     'L-16');
insert into borrower values('Curry',     'L-93');
insert into borrower values('Hayes',     'L-15');
insert into borrower values('Jackson',   'L-14');
```

insert into borrower values('Jones', 'L-17');
insert into borrower values('Smith', 'L-11');
insert into borrower values('Smith', 'L-23');
insert into borrower values('Williams', 'L-17');

insert into employee values('Adams', 'Perryridge', 1500);
insert into employee values('Brown', 'Perryridge', 1300);
insert into employee values('Gopal', 'Perryridge', 5300);
insert into employee values('Johnson', 'Downtown', 1500);
insert into employee values('Loreena', 'Downtown', 1300);
insert into employee values('Peterson', 'Downtown', 2500);
insert into employee values('Rao', 'Austin', 1500);
insert into employee values('Sato', 'Austin', 1600);

You can use SQL command 'select * from [table_name];' to check if data have been inserted.

(4). Perform queries on those tables

1. Find all account whose balance is smaller than 500.

Answer: *select account_name from account where balance < 500;*

2. Find all name of customers whose city is in Brooklyn

Answer: *select customer_name from customer where customer_city='Brooklyn';*

3. Find all employees whose salary is greater than 1400 and working branch is not 'Downtown'

Answer: *select * from employee where salary>1400 and branch_name<>'Downtown';*

4. Calculate the average salary of all employees and show the average salary as "avg_salary"

Answer: *select avg(salary) as avg_salary from employee* ;

5. Calculate the number of customer for each account

Answer: *select account_number, count(distinct customer_name) from depositor group by account_number;*

6. Show all account_number, branch_name and corresponding branch_city

Answer: *select account_number, branch.branch_name, branch_city from account, branch where account.branch_name=branch.branch_name;*

Questions:

1. Find the names of all customers.
2. Find the names of all branches in the loan relation, don't display duplicates.
3. Display the entire Branch table.
4. Find the account number for all accounts where the balance is greater than $700.
5. Find the account number and balance for all accounts from Brighton where the balance is greater than $800.
6. Display the branch name and assets from all branches in thousands of dollars and rename the assets column to 'assets in thousands'.
7. Find the name of all branches with assets between one and four million dollars.
8. Find the name, account number, and balance of all customers who have an account.
9. Find the name, account number, and balance of all customers who have an account with a balance of $400 or less.

--**SOLUTION**##
```
select customer_name
from   Customer
;

-------------------------------------------------
select distinct branch_name
from   loan
;

-------------------------------------------------
select *
from Branch
;

-------------------------------------------------
select account_number from account
where  balance > 700
;

-------------------------------------------------
select account_number, balance from account
where  balance > 800 and branch_name = 'Brighton'
;

-------------------------------------------------
select branch_name, (assets / 1000) as 'assets in thousands'
from branch
;

-------------------------------------------------
select branch_name
from branch
where assets between 1000000 and 4000000
;

-------------------------------------------------
select customer_name, borrower.loan_number, amount
from   borrower, loan
where  borrower.loan_number = loan.loan_number  and
       branch_name = 'Perryridge';

select depositor.customer_name, account.account_number, balance
from   depositor, account
where  depositor.account_number = account.account_number
;

-------------------------------------------------
select depositor.customer_name, account.account_number, balance
from   depositor, account
where  depositor.account_number = account.account_number
       and balance <= 400
;
```