bank Database

Instances

Branch

branch_name	branch_city	assets
Brighton	Brooklyn	7100000
Downtown	Brooklyn	9000000
Mianus	Horseneck	400000
North Town	Rye	3700000
Perryridge	Horseneck	1700000
Pownal	Bennington	300000
Redwood	Palo Alto	2100000
Round Hill	Horseneck	8000000

Customer

customer_name	customer_street	customer_city
Adams	Spring	Pittsfield
Brooks	Senator	Brooklyn
Curry	North	Rye
Glenn	Sand Hill	Woodside
Green	Walnut	Stamford
Hayes	Main	Harrison
Johnson	Alma	Palo Alto
Jones	Main	Harrison
Linsday	Park	Pittsfield
Smith	North	Rye
Turner	Putnam	Stamford
Williams	Nassau	Princeton

Account

account_number	branch_name	balance
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700
A-305	Round Hill	350

Loan

loan_number	branch_name	amount
L-11	Round Hill	900
L-14	Downtown	1500
L-15	Perryridge	1500
L-16	Perryridge	1300
L-17	Downtown	1000
L-23	Redwood	2000
L-93	Mianus	500

Depositor

customer_name	account_number
Hayes	A-102
Johnson	A-101
Johnson	A-201
Jones	A-217
Linsday	A-222
Smith	A-215
Turner	A-305

Borrower

customer_name	loan_number
Adams	L-16
Curry	L-93
Hayes	L-15
Jackson	L-14
Jones	L-17
Smith	L-11
Smith	L-23
Williams	L-17

Creating branch table

```sql
create table branch(
    branch_name varchar2(15) constraint branch_name_pr primary key,
    branch_city varchar2(15),
    assets int);
```

```
DESC branch;

+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| branch_name | varchar(15) | NO   | PRI | NULL    |       |
| branch_city | varchar(15) | YES  |     | NULL    |       |
| assets      | int(11)     | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
3 rows in set (0.001 sec)
```

Creating customer table

```
create table customer(
    customer_name varchar2(15) constraint customer_name_pr primary key,
    customer_street varchar2(15),
    customer_city varchar2(15));
```

```
DESC customer;

+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| customer_name   | varchar(15) | NO   | PRI | NULL    |       |
| customer_street | varchar(15) | YES  |     | NULL    |       |
| customer_city   | varchar(15) | YES  |     | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
3 rows in set (0.001 sec)
```

The above two tables are the master tables

Creating account table

```
create table account(
    account_number varchar2(6),
    branch_name varchar2(15),
    balance int,
    constraint accnum_branchname_pk primary key(account_number, branch_name),
    foreign key (branch_name) references branch(branch_name));
```

```
DESC account;

+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| account_number | varchar(6)  | NO   | PRI | NULL    |       |
| branch_name    | varchar(15) | NO   | PRI | NULL    |       |
| balance        | int(11)     | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
3 rows in set (0.001 sec)
```

Creating depositor table

```
create table depositor(
    customer_name varchar2(15),
    account_number varchar2(6),
    constraint cname_anum_pr primary key(customer_name, account_number),
    foreign key (customer_name) references customer(customer_name),
    foreign key (account_number) references account(account_number));
```

```
DESC depositor;
```

```
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| customer_name  | varchar(15) | NO   | PRI | NULL    |       |
| account_number | varchar(6)  | NO   | PRI | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
2 rows in set (0.001 sec)
```

Creating loan table

```
create table loan(
    loan_number varchar2(5),
    branch_name varchar2(15),
    amount int,
    constraint lnum_bname_pr primary key(loan_number, branch_name),
    foreign key (branch_name) references branch(branch_name));
```

```
DESC loan;
```

```
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| loan_number | varchar(5)  | NO   | PRI | NULL    |       |
| branch_name | varchar(15) | NO   | PRI | NULL    |       |
| amount      | int(11)     | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
3 rows in set (0.000 sec)
```

Creating borrower table

```
create table borrower(
    customer_name varchar2(15),
    loan_number varchar2(5),
    constraint cname_lnum_pr primary key(customer_name, loan_number ),
    foreign key (customer_name) references customer(customer_name),
    foreign key (loan_number) references loan(loan_number));

DESC borrower;
```

```
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| customer_name | varchar(15) | NO   | PRI | NULL    |       |
| loan_number   | varchar(5)  | NO   | PRI | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
2 rows in set (0.000 sec)
```

Inserting Data

```
INSERT INTO branch (branch_name, branch_city, assets) VALUES
('Brighton', 'Brooklyn', 7100000),
('Downtown', 'Brooklyn', 9000000),
('Mianus', 'Horseneck', 400000),
('North Town', 'Rye', 3700000),
('Perryridge', 'Horseneck', 1700000),
('Pownal', 'Bennington', 300000),
('Redwood', 'Palo Alto', 2100000),
('Round Hill', 'Horseneck', 8000000);

INSERT INTO customer (customer_name, customer_street, customer_city) VALUES
('Adams', 'Spring', 'Pittsfield'),
('Brooks', 'Senator', 'Brooklyn'),
('Curry', 'North', 'Rye'),
('Glenn', 'Sand Hill', 'Woodside'),
('Green', 'Walnut', 'Stamford'),
('Hayes', 'Main', 'Harrison'),
('Johnson', 'Alma', 'Palo Alto'),
('Jones', 'Main', 'Harrison'),
('Linsday', 'Park', 'Pittsfield'),
('Smith', 'North', 'Rye'),
('Turner', 'Putnam', 'Stamford'),
('Williams', 'Nassau', 'Princeton');

INSERT INTO account (account_number, branch_name, balance) VALUES
('A-101', 'Downtown', 500),
('A-102', 'Perryridge', 400),
('A-201', 'Brighton', 900),
('A-215', 'Mianus', 700),
('A-217', 'Brighton', 750),
('A-222', 'Redwood', 700),
('A-305', 'Round Hill', 350);

INSERT INTO loan (loan_number, branch_name, amount) VALUES
('L-11', 'Round Hill', 900),
('L-14', 'Downtown', 1500),
('L-15', 'Perryridge', 1500),
('L-16', 'Perryridge', 1300),
('L-17', 'Downtown', 1000),
('L-23', 'Redwood', 2000),
('L-93', 'Mianus', 500);


INSERT INTO depositor (customer_name , account_number ) VALUES
('Hayes','A-102'),
('Johnson', 'A-101'),
('Johnson', 'A-201'),
('Jones', 'A-217'),
('Linsday', 'A-222'),
('Smith', 'A-215'),
('Turner', 'A-305');

INSERT INTO borrower (customer_name , loan_number ) VALUES
('Adams','L-16'),
('Curry','L-93'),
('Hayes','L-15'),
('Jackson','L-14'),
('Jones','L-17'),
('Smith','L-11'),
('Smith','L-23'),
('Williams','L-17');
```

Database Excercises

1 - Retrieve all data from customer, branch, depositor, loan, account, borrower

```
SELECT * FROM branch;
SELECT * FROM customer;
SELECT * FROM depositor;
SELECT * FROM loan;
SELECT * FROM account;
SELECT * FROM borrower;
```

2 - Retrieve the names and cities of all borrowers

```
SELECT customer_name, customer_city
FROM customer
WHERE customer_name in (SELECT customer_name FROM borrower);
```

3 - Retrieve set of names and cities of customers who have loan at ‘Perryridge’ branch

```
SELECT borrower.customer_name, branch.branch_city
from borrower
JOIN loan
on loan.loan_number = borrower.loan_number
JOIN branch
on loan.branch_name = 'Perryridge' AND branch.branch_name = 'Perryridge';
```

4 - Retrieve the number of accounts with balance between 700 and 900

```
SELECT count(account_number)
FROM account
WHERE balance BETWEEN 700 AND 900;
```

5 - Retrieve the names of customer on streets with names ending in ‘hill’ - string pattern matching

```
SELECT customer_name
FROM customer
WHERE customer_street like '%hill';
```

6 - Retrieve the names of customer with both account and loan at ‘Perryridge’

```
SELECT depositor.customer_name
FROM depositor
JOIN borrower
on borrower.customer_name = depositor.customer_name
JOIN loan
on loan.branch_name = 'Perryridge' and loan.loan_number = borrower.loan_number
JOIN account
on account.branch_name = 'Perryridge' and account.account_number = depositor.account_number
```

7 - Retrieve the names of customer with account but not a loan at ‘Perryridge’

```
SELECT depositor.customer_name
FROM depositor
JOIN borrower
on borrower.customer_name = depositor.customer_name
```

```
JOIN loan
on loan.branch_name != 'Perryridge' and loan.loan_number = borrower.loan_number
JOIN account
on account.branch_name = 'Perryridge' and account.account_number = depositor.account_number;
```

8 - List the name and cities of all borrowers

```
select borrower.customer_name, branch.branch_city
FROM borrower
LEFT JOIN loan
on borrower.loan_number = loan.loan_number
JOIN branch
on branch.branch_name = loan.branch_name;
```

9 - Retrieve the set of names of customers where accounts at a branch ‘Hayes’ has

```
SELECT depositor.customer_name
FROM account
JOIN depositor
on depositor.account_number = account.account_number
WHERE depositor.customer_name = 'Hayes';
```

10 - Retrieve the set of names of branches having largest average balance

```
SELECT max(avg_balance)
FROM (
    SELECT branch_name,avg(balance) as avg_balance
    from account
    GROUP by branch_name);
```

11 - Retrieve the whose assets are greater than the assets of some branch in brooklyn.

```
SELECT branch.branch_name
FROM branch
WHERE assets > any(
    SELECT assets
    FROM branch
    WHERE branch_city = 'brooklyn');
```

12 - Retrieve the names of customers with both account and loans at ‘Perryridge’ branch.

```
SELECT customer.customer_name
FROM customer
```

```
LEFT JOIN depositor
on depositor.customer_name = customer.customer_name
LEFT JOIN borrower
on borrower.customer_name = customer.customer_name
LEFT JOIN account
on account.account_number = depositor.account_number
LEFT JOIN loan
on loan.loan_number = borrower.loan_number
WHERE loan.branch_name = 'Perryridge' OR account.branch_name = 'Perryridge'
```

13 - Retrieve the names of customer at 'Perryridge' branch in alphabetical order.

```
SELECT customer.customer_name
FROM customer
LEFT JOIN depositor
on depositor.customer_name = customer.customer_name
LEFT JOIN borrower
on borrower.customer_name = customer.customer_name
LEFT JOIN account
on account.account_number = depositor.account_number
LEFT JOIN loan
on loan.loan_number = borrower.loan_number
WHERE loan.branch_name = 'Perryridge' or account.branch_name = 'Perryridge'
ORDER by customer.customer_name ASC
```

14 - Retrieve the loan data order by decreasing amonuts and than increasing loan numbers

```
SELECT * FROM loan
ORDER by amount DESC , loan_number;
```

15 - Retrieve the names of branches having atleast 1 account having average balance.

16 - Retrieve the names of branches having atleast one account, with size of set of customers having one account one account at that branch

17 - Print average balance of all accounts.

```
SELECT avg(balance) FROM account;
```

18 - Find the names of branches having atleast 1 account with average balances of accounts at each branch, if that balance is above 700

19 - Find a name or names of branch / branches having largest average balance.

```
SELECT max(avg_balance)
FROM (
    SELECT branch_name,avg(balance) as avg_balance
    from account
    GROUP by branch_name);
```

20 - Find the number of customers

```
SELECT count(customer_name) from customer;
```

21 - Find average balance of all customers in ‘Harrison’, having atleast 2 accounts.

```
SELECT avg(balance)
FROM account
JOIN depositor
on depositor.account_number = account.account_number
JOIN customer
on customer.customer_name = depositor.customer_name
where (SELECT count(customer_name) FROM depositor) >= 2 AND customer.customer_city = 'Harris
```