Name: Sudhir Singh
Class: TE - A
Batch: A3
Roll No: 23152
Subject: SL-I (A)

Lab Assignment 1

Problem Statement: Account(Acc_no, branch_name,balance)
branch(branch_name,branch_city,assets) customer(cust_name,cust_street,cust_city)
Depositor(cust_name,acc_no) Loan(loan_no,branch_name,amount) Borrower(cust_name,loan_no)
Solve following query:
Create above tables with appropriate constraints like primary key, foreign key, check constrains, not null etc.

```
mysql> CREATE DATABASE Bank;
Query OK, 1 row affected (0.01 sec)

mysql> use bank;
Database changed
mysql> CREATE TABLE Branch (
    ->   branch_name VARCHAR(50) PRIMARY KEY,
    ->   branch_city VARCHAR(50) NOT NULL,
    ->   assets DECIMAL(15,2) CHECK (assets >= 0)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> CREATE TABLE Customer (
    ->   cust_name VARCHAR(50) PRIMARY KEY,
    ->   cust_street VARCHAR(100) NOT NULL,
    ->   cust_city VARCHAR(50) NOT NULL
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE Account (acc_no INT PRIMARY KEY,
    ->   branch_name VARCHAR(50) NOT NULL,
    ->   balance DECIMAL(12,2) CHECK (balance >= 0),
    ->   FOREIGN KEY (branch_name) REFERENCES Branch(branch_name)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE Depositor (
    ->   cust_name VARCHAR(50),
    ->   acc_no INT,
    ->   PRIMARY KEY (cust_name, acc_no),
    ->   FOREIGN KEY (cust_name) REFERENCES Customer(cust_name),
    ->   FOREIGN KEY (acc_no) REFERENCES Account(acc_no)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> CREATE TABLE Loan (
    ->   loan_no INT PRIMARY KEY,
    ->   branch_name VARCHAR(50) NOT NULL,
    ->   amount DECIMAL(12,2) CHECK (amount > 0),
    ->   FOREIGN KEY (branch_name) REFERENCES Branch(branch_name)
    -> );
```

```
mysql> CREATE TABLE Borrower (
    ->   cust_name VARCHAR(50),
    ->   loan_no INT,
    ->   PRIMARY KEY (cust_name, loan_no),
    ->   FOREIGN KEY (cust_name) REFERENCES Customer(cust_name),
    ->   FOREIGN KEY (loan_no) REFERENCES Loan(loan_no)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> INSERT INTO Branch VALUES ('Akurdi','Pune',500000);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Branch VALUES ('Nigdi','Pune',400000);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Account VALUES (101,'Akurdi',15000);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Account VALUES (102,'Akurdi',8000);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Account VALUES (103,'Nigdi',12000);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Customer VALUES ('Ravi','MG Road','Pune');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Customer VALUES ('Anita','FC Road','Pune');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Customer VALUES ('Sanjay','Main Street','Mumbai');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Depositor VALUES ('Ravi',101);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Depositor VALUES ('Anita',102);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Loan VALUES (201,'Akurdi',15000);
```

```
mysql> INSERT INTO Loan VALUES (201,'Akurdi',15000);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Loan VALUES (202,'Akurdi',10000);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Loan VALUES (203,'Nigdi',18000);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Borrower VALUES ('Ravi',201);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Borrower VALUES ('Sanjay',203);
Query OK, 1 row affected (0.01 sec)

mysql> SHOW TABLES;
+----------------+
| Tables_in_bank |
+----------------+
| account        |
| borrower       |
| branch         |
| customer       |
| depositor      |
| loan           |
+----------------+
6 rows in set (0.01 sec)

mysql> Select * from account;
+--------+-------------+----------+
| acc_no | branch_name | balance  |
+--------+-------------+----------+
|    101 | Akurdi      | 15000.00 |
|    102 | Akurdi      |  8000.00 |
|    103 | Nigdi       | 12000.00 |
+--------+-------------+----------+
3 rows in set (0.00 sec)
```

```
mysql> Select * from borrower;
+-----------+---------+
| cust_name | loan_no |
+-----------+---------+
| Ravi      |     201 |
| Sanjay    |     203 |
+-----------+---------+
2 rows in set (0.00 sec)

mysql> Select * from branch;
+-------------+-------------+-----------+
| branch_name | branch_city | assets    |
+-------------+-------------+-----------+
| Akurdi      | Pune        | 500000.00 |
| Nigdi       | Pune        | 400000.00 |
+-------------+-------------+-----------+
2 rows in set (0.00 sec)

mysql> Select * from customer;
+-----------+-------------+-----------+
| cust_name | cust_street | cust_city |
+-----------+-------------+-----------+
| Anita     | FC Road     | Pune      |
| Ravi      | MG Road     | Pune      |
| Sanjay    | Main Street | Mumbai    |
+-----------+-------------+-----------+
3 rows in set (0.00 sec)

mysql> Select * from despositor;
ERROR 1146 (42S02): Table 'bank.despositor' doesn't exist
mysql> Select * from depositor;
+-----------+--------+
| cust_name | acc_no |
+-----------+--------+
| Ravi      |    101 |
| Anita     |    102 |
+-----------+--------+
2 rows in set (0.00 sec)
mysql> Select * from loan;
+---------+-------------+----------+
| loan_no | branch_name | amount   |
+---------+-------------+----------+
|     201 | Akurdi      | 15000.00 |
|     202 | Akurdi      | 10000.00 |
|     203 | Nigdi       | 18000.00 |
+---------+-------------+----------+
3 rows in set (0.00 sec)
```

Q1. Find the names of all branches in loan relation.

```
mysql> -- Q1. Find the names of all branches in loan relation.
mysql> SELECT DISTINCT branch_name FROM Loan;
+-------------+
| branch_name |
+-------------+
| Akurdi      |
| Nigdi       |
+-------------+
2 rows in set (0.00 sec)
```

Q2. Find all loan numbers for loans made at Akurdi Branch with loan amount > 12000.

```
mysql> -- Q2 Loan numbers at Akurdi with amount > 12000
mysql> SELECT loan_no FROM Loan WHERE branch_name='Akurdi' AND amount>12000;
+---------+
| loan_no |
+---------+
|     201 |
+---------+
1 row in set (0.01 sec)
```

Q3. Find all customers who have a loan from bank. Find their names, loan_no and loan amount.

```
mysql> -- Q3 Customers with loan (name, loan_no, amount)
mysql> SELECT B.cust_name, L.loan_no, L.amount FROM Borrower B JOIN Loan L ON B.loan_no=L.loan_no;
+-----------+---------+----------+
| cust_name | loan_no | amount   |
+-----------+---------+----------+
| Ravi      |     201 | 15000.00 |
| Sanjay    |     203 | 18000.00 |
+-----------+---------+----------+
2 rows in set (0.00 sec)
```

Q4. List all customers in alphabetical order who have loan from Akurdi branch.

```
mysql> -- Q4 Customers with loan from Akurdi in alphabetical order
mysql> SELECT DISTINCT B.cust_name FROM Borrower B JOIN Loan L ON B.loan_no=L.loan_no WHERE L.branch_name='Akurdi' ORDER BY B.cust_name;
+-----------+
| cust_name |
+-----------+
| Ravi      |
+-----------+
1 row in set (0.01 sec)
```

Q5. Find all customers who have an account or loan or both at bank.

```
mysql> -- Q5 Customers who have account or loan or both
mysql> SELECT cust_name FROM Depositor UNION SELECT cust_name FROM Borrower;
+-----------+
| cust_name |
+-----------+
| Ravi      |
| Anita     |
| Sanjay    |
+-----------+
3 rows in set (0.00 sec)
```

Q6. Find all customers who have both account and loan at bank.

```
mysql> -- Q6 Customers with both account and loan
mysql> SELECT DISTINCT D.cust_name FROM Depositor D JOIN Borrower B ON D.cust_name=B.cust_name;
+-----------+
| cust_name |
+-----------+
| Ravi      |
+-----------+
1 row in set (0.00 sec)
```

Q7. Find all customer who have account but no loan at the bank.

```
mysql> -- Q7 Customers with account but no loan
mysql> SELECT cust_name FROM Depositor WHERE cust_name NOT IN (SELECT cust_name FROM Borrower);
+-----------+
| cust_name |
+-----------+
| Anita     |
+-----------+
1 row in set (0.00 sec)
```

Q8. Find average account balance at Akurdi branch.

```
mysql> -- Q8 Average account balance at Akurdi
mysql> SELECT AVG(balance) AS avg_balance FROM Account WHERE branch_name='Akurdi';
+--------------+
| avg_balance  |
+--------------+
| 11500.000000 |
+--------------+
1 row in set (0.01 sec)
```

Q9. Find the average account balance at each branch.

```
mysql> -- Q9 Average account balance at each branch
mysql> SELECT branch_name, AVG(balance) AS avg_balance FROM Account GROUP BY branch_name;
+-------------+--------------+
| branch_name | avg_balance  |
+-------------+--------------+
| Akurdi      | 11500.000000 |
| Nigdi       | 12000.000000 |
+-------------+--------------+
2 rows in set (0.00 sec)
```

Q10. Find no. of depositors at each branch.

```
mysql> -- Q10 No. of depositors at each branch
mysql> SELECT A.branch_name, COUNT(DISTINCT D.cust_name) AS no_of_depositors FROM Account A JOIN Depositor D ON A.acc_no=D.acc_no GROUP BY A.branch_name;
+-------------+-----------------+
| branch_name | no_of_depositors |
+-------------+-----------------+
| Akurdi      |               2 |
+-------------+-----------------+
1 row in set (0.01 sec)
```

Q11. Find the branches where average account balance > 12000.

```
mysql> -- Q11 Branches where avg balance > 12000
mysql> SELECT branch_name FROM Account GROUP BY branch_name HAVING AVG(balance)>12000;
Empty set (0.00 sec)
```

Q12. Find number of tuples in customer relation.

```
mysql> -- Q12 Number of tuples in customer
mysql> SELECT COUNT(*) AS total_customers FROM Customer;
+-----------------+
| total_customers |
+-----------------+
|               3 |
+-----------------+
1 row in set (0.01 sec)
```

Q13. Calculate total loan amount given by bank.

```
mysql> -- Q13 Total loan amount given by bank
mysql> SELECT SUM(amount) AS total_loan FROM Loan;
+------------+
| total_loan |
+------------+
|   43000.00 |
+------------+
1 row in set (0.00 sec)
```

Q14. Delete all loans with loan amount between 1300 and 1500.

```
mysql> -- Q14 Delete all loans with amount between 1300 & 1500
mysql> DELETE FROM Loan WHERE amount BETWEEN 1300 AND 1500;
Query OK, 0 rows affected (0.00 sec)
```

Q15. Delete all tuples at every branch located in Nigdi.

```
mysql> -- Q15 Delete all tuples at branches located in Nigdi
mysql> DELETE FROM Branch WHERE branch_city='Nigdi';
Query OK, 0 rows affected (0.00 sec)
```

Q16. Create synonym for customer table as cust.

```
mysql> -- Q16 Create synonym for customer as cust
mysql> SELECT * FROM Customer AS cust;
+-----------+-------------+-----------+
| cust_name | cust_street | cust_city |
+-----------+-------------+-----------+
| Anita     | FC Road     | Pune      |
| Ravi      | MG Road     | Pune      |
| Sanjay    | Main Street | Mumbai    |
+-----------+-------------+-----------+
3 rows in set (0.00 sec)
```

Q17. Create sequence roll_seq and use in student table for roll_no column.

```
mysql> -- Q17 Create sequence roll_seq & use in student table
mysql> CREATE TABLE Student (
    ->   roll_no INT AUTO_INCREMENT PRIMARY KEY,
    ->   name VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO Student (name) VALUES ('Ravi');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Student (name) VALUES ('Anita');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Student (name) VALUES ('Sanjay');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM Student;
+---------+--------+
| roll_no | name   |
+---------+--------+
|       1 | Ravi   |
|       2 | Anita  |
|       3 | Sanjay |
+---------+--------+
3 rows in set (0.00 sec)

mysql> Sudhir singh TEAD23152
```