ASSIGNMENT NO.1

```
CREATE TABLE customer(
  cust_name varchar(20) PRIMARY KEY,
  cust_street varchar(20),
  cust_city varchar(20)
);
CREATE TABLE branch(
  branch_name varchar(20) PRIMARY KEY,
  branch_city varchar(20),
  assets int
);
CREATE TABLE Accounts(
  Acc_no int PRIMARY KEY,
  branch_name varchar(20),
  balance int,
  FOREIGN KEY (branch_name) REFERENCES branch(branch_name) ON DELETE CASCADE
);
CREATE TABLE Depositor(
  cust_name varchar(20),
  acc_no int,
  FOREIGN KEY (Acc_no) REFERENCES Accounts(Acc_no) ON DELETE CASCADE,
  FOREIGN KEY (cust_name) REFERENCES customer(cust_name) ON DELETE CASCADE
);
CREATE TABLE Loan(
  loan_no int PRIMARY KEY,
  branch_name varchar(20),
  amount float,
  FOREIGN KEY (branch_name) REFERENCES branch(branch_name) ON DELETE CASCADE
);
```

```
CREATE TABLE Borrower(
  cust_name varchar(20),
  loan_no int,
  FOREIGN KEY (loan_no) REFERENCES Loan(loan_no) ON DELETE CASCADE
);
INSERT INTO Branch VALUES('SBI Pen','Pen',10000000);
INSERT INTO Branch VALUES('SBI Alibag','Alibag',9000000);
INSERT INTO Branch VALUES('SBI Akurdi','Akurdi',99000000);
INSERT INTO Branch VALUES('SBI Thane','Thane',75000000);
INSERT INTO Branch VALUES('SBI Kalyan','Kalyan',60000000);
INSERT INTO Branch VALUES('SBI Borivali','Borivali',85000000);
INSERT INTO Accounts VALUES ('1','SBI Pen',5000);
INSERT INTO Accounts VALUES ('2','SBI Pen',20000);
INSERT INTO Accounts VALUES ('3','SBI Alibag',40000);
INSERT INTO Accounts VALUES ('4','SBI Thane',30000);
INSERT INTO Accounts VALUES ('5','SBI Kalyan',25000);
INSERT INTO Accounts VALUES ('6','SBI Borivali',400);
INSERT INTO Customer VALUES ('Soham','LBS Marg','Pen');
INSERT INTO Customer VALUES ('Atharva','LT Marg','Pen');
INSERT INTO Customer VALUES ('Vedant','Near Pccoe','Akurdi');
INSERT INTO Customer VALUES ('Ramesh','Powai','Thane');
INSERT INTO Customer VALUES ('Suresh','Vikhroli','Kalyan');
INSERT INTO Customer VALUES ('Geeta','Goregaon','Borivali');
INSERT INTO Loan VALUES ('11','SBI Pen',1500000);
INSERT INTO Loan VALUES ('12','SBI Alibag',3000000);
INSERT INTO Loan VALUES ('13','SBI Akurdi',9000000);
INSERT INTO Loan VALUES ('14','SBI Thane',2000000);
INSERT INTO Loan VALUES ('15','SBI Kalyan',5000000);
INSERT INTO Loan VALUES ('16','SBI Borivali',7000000)
INSERT INTO Borrower VALUES ('Soham',11);
INSERT INTO Borrower VALUES ('Atharva',12);
INSERT INTO Borrower VALUES ('Vedant',13);
```

```
INSERT INTO Borrower VALUES ('Ramesh',14);

INSERT INTO Borrower VALUES ('Suresh',15);

INSERT INTO Borrower VALUES ('Geeta',16);

INSERT INTO Depositor VALUES ('Soham',1);

INSERT INTO Depositor VALUES ('Atharva',2);

INSERT INTO Depositor VALUES ('Vedant',3);

INSERT INTO Depositor VALUES ('Ramesh',4);

INSERT INTO Depositor VALUES ('Suresh',5);

INSERT INTO Depositor VALUES ('Geeta',6);
SQL> --1.Find the names of all branches in loan relation.

SQL> SELECT DISTINCT branch_name FROM Loan;

BRANCH_NAME

--------------------

SBI Alibag

SBI Thane

SBI Akurdi

SBI Kalyan

SBI Pen
SQL> --2. Find all loan numbers for loans made at Akurdi Branch with loan amount >12000.

SQL>  SELECT LOAN_NO FROM LOAN WHERE AMOUNT>12000 AND BRANCH_NAME='SBI Akurdi';


  LOAN_NO

----------

      13
SQL> --3. Find no. of depositors at each branch.

SQL> SELECT COUNT(*) FROM DEPOSITOR;

 COUNT(*)

----------

       6
SQL> --4. Delete all loans with loan amount between 1300 and 1500.

SQL>  DELETE FROM LOAN WHERE AMOUNT>1300 AND AMOUNT <1500;

0 rows deleted.
```

```
SQL> --5. Delete all tuples at every branch located in Nigdi.

SQL>  DELETE FROM Accounts WHERE BRANCH_NAME='SBI Akurdi';

0 rows deleted.

SQL> --6. Delete all account tuples at every branch located in a specific city.

SQL> DELETE FROM Accounts WHERE branch_name IN (SELECT branch_name FROM branch WHERE branch_city =
'Alibag');

1 row deleted.

SQL> --8. Find the names of all customers who have taken loans.

SQL> SELECT CUST_NAME FROM Borrower LEFT JOIN Loan ON Borrower.loan_no = Loan.loan_no;

CUST_NAME

--------------------

Atharva

Vedant

Ramesh

Suresh

SQL> --9. Find the names of all customers who have not taken loans.

SQL> SELECT CUST_NAME FROM Borrower FULL OUTER JOIN Loan ON Borrower.loan_no = Loan.loan_no WHERE
Borrower.loan_no IS NULL;

CUST_NAME

--------------------

SQL> --10. Find the name, account number, and balance of all customers who have an account with account balance
of 400 or less.

SQL> SELECT customer.cust_name, Accounts.Acc_no, Accounts.balance

  2  FROM customer

  3  JOIN Depositor ON customer.cust_name = Depositor.cust_name

  4  JOIN Accounts ON Depositor.acc_no = Accounts.Acc_no WHERE Accounts.balance<400;

no rows selected

SQL> --11. Find the name, account number, and balance of all customers who have an account.

SQL> SELECT customer.cust_name, Accounts.Acc_no, Accounts.balance

  2  FROM customer

  3  JOIN Depositor ON customer.cust_name = Depositor.cust_name

  4  JOIN Accounts ON Depositor.acc_no = Accounts.Acc_no;

CUST_NAME              ACC_NO    BALANCE
```

---------- ---------- ----------

Soham 1 5000

Atharva 2 20000

Ramesh 4 30000

Suresh 5 25000

Geeta 6 400

SQL> --12. Find the name of all branches with assets between 10K AND 50K.

SQL> SELECT SUM(Accounts.Balance), Accounts.branch_name FROM Accounts JOIN Branch ON Accounts.branch_name = Branch.branch_name GROUP BY Accounts.branch_name;

SUM(ACCOUNTS.BALANCE) BRANCH_NAME

---------- ----------

30000 SBI Thane

400 SBI Borivali

25000 SBI Kalyan

25000 SBI Pen

SQL> --15. Drop table Depositor.

SQL> DROP TABLE DEPOSITOR;

Table dropped

SQL> --16. Truncate table Borrower.

SQL> TRUNCATE TABLE BORROWER;

Table truncated.

SQL> --Q.2 Create table college (college_id primary key, college_code, college-name)

SQL> CREATE TABLE college(

2 college_id int PRIMARY KEY,

3 college_code int,

4 college_name varchar(10)

5);

Table created.

SQL> --1. Create Index College_Index using using any column.

SQL> CREATE INDEX CLG_NM ON college(college_name);

Index created.

SQL> --2. Create unique index for unique values.

```
SQL>  CREATE UNIQUE INDEX CLG_ID ON college(college_code);

Index created.

SQL> --Q.3 Create synonym for customer table as cust.

SQL> CREATE SYNONYM cust FOR customer;

Synonym created.

SQL> --Q.4 Create sequence roll_seq and use in college table for roll_no column.

SQL> CREATE SEQUENCE rol_seq
  2     START WITH 1
  3     INCREMENT BY 1;

Sequence created.
```