HUAWEI INTERNSHIP
ASSIGNMENT – PHASE 1

NAME: Yalini B **DATE**: 14.03.2022
COLLEGE: Coimbatore Institute of Technology

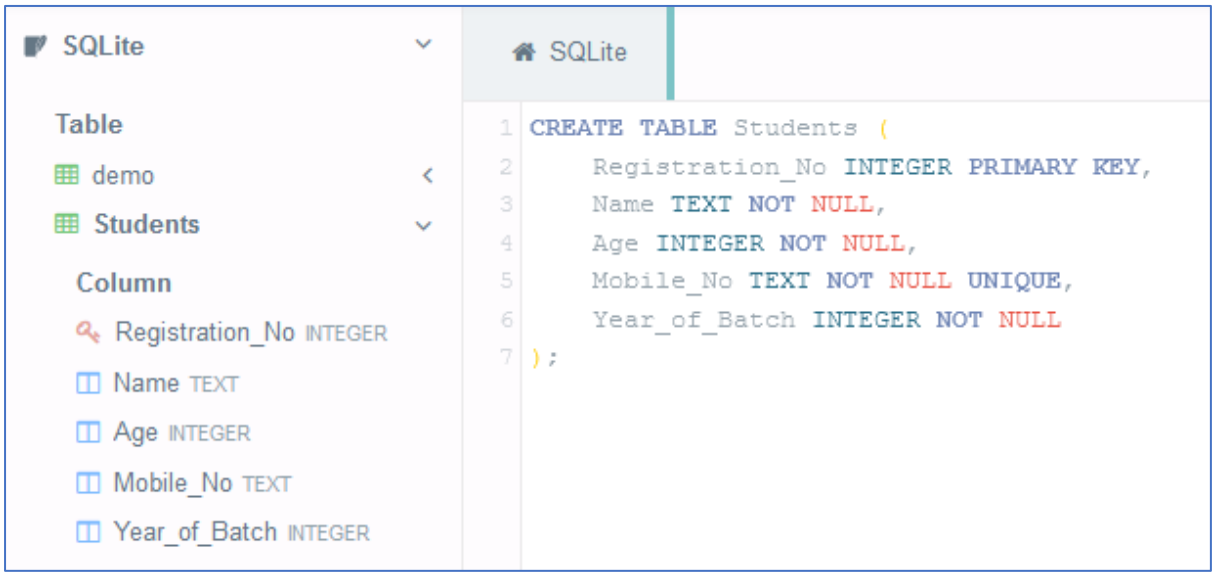
ASSIGNMENT 1

Create a DB for your college with following parameters:

1. Create a table for students with name, age, mobile no, registration no., year of batch as columns.
2. Create a table for teachers with name, domain, department as column
3. Write a query to insert 10 students data and 10 teachers data in the respective table
4. Write a query to fetch all the students from 2020 batch
5. Write a query to fetch all teachers from CS department
6. Write a query to edit at least 3 records of students
7. Write a query to delete 2 records from teachers table

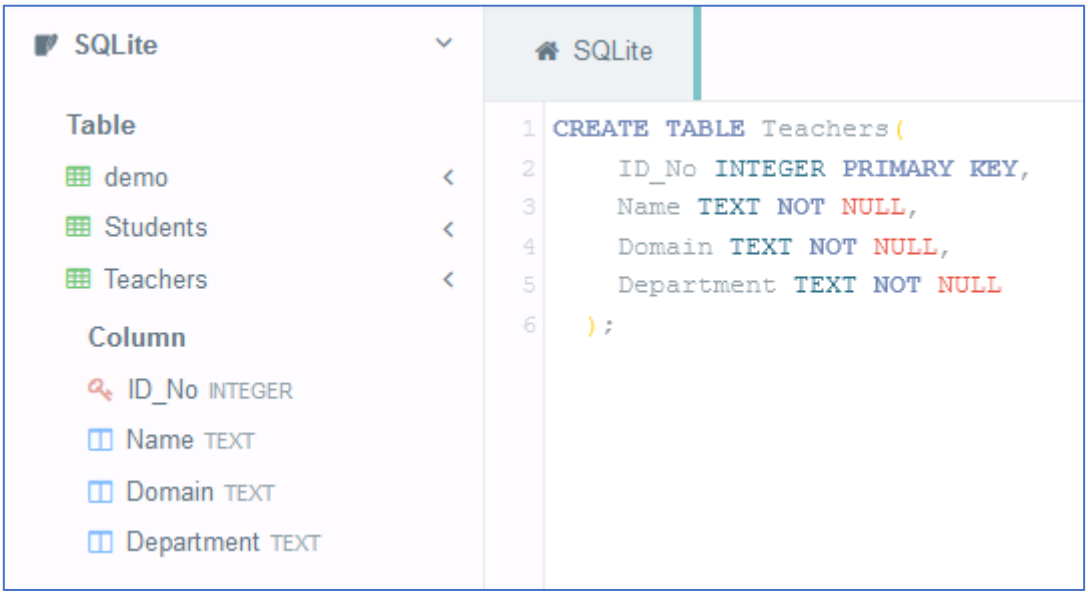
1) Creating a table for students:

```
CREATE TABLE Students (
        Registration_No INTEGER PRIMARY KEY,
        Name TEXT NOT NULL,
        Age INTEGER NOT NULL,
        Mobile_No TEXT NOT NULL UNIQUE,
        Year_of_Batch INTEGER NOT NULL
);
```



2) Creating a table for teachers:

```
CREATE TABLE Teachers(
   ID_No INTEGER PRIMARY KEY,
   Name TEXT NOT NULL,
   Domain TEXT NOT NULL,
   Department TEXT NOT NULL
 );
```



3a) Inserting 10 students data

```
INSERT INTO Students
        (Registration_No, Name, Age, Mobile_No, Year_of_Batch)
VALUES
        (1807001, "Aditya", 21, 9998978670, 2018),
   (1901001, "Viran", 20, 7734978020, 2019),
   (1903023, "Kamali", 21, 7834895674, 2019),
   (2004041, "Kala", 20, 9887956402, 2020),
   (2007035, "Janani", 19, 8834512390, 2020),
   (1802001, "Malathi", 21, 7845678670, 2018),
   (1905031, "Tamil", 21, 7869023456, 2018),
   (2007078, "Ezhil", 21, 7789345609, 2020),
   (2107001, "Anaya", 19, 8968978670, 2021),
   (2007056, "Yalini", 20, 9034678295, 2020)
```

SQLite

```
1 SELECT * FROM Students;
```

Registration_No	Name	Age	Mobile_No	Year_of_Batch
1802001	Malathi	21	7845678670	2018
1807001	Aditya	21	9998978670	2018
1901001	Viran	20	7734978020	2019
1903023	Kamali	21	7834895674	2019
1905031	Tamil	21	7869023456	2018
2004041	Kala	20	9887956402	2020
2007035	Janani	19	8834512390	2020
2007056	Yalini	20	9034678295	2020
2007078	Ezhil	21	7789345609	2020
2107001	Anaya	19	8968978670	2021

3b) Inserting 10 Teachers data

```
INSERT INTO Teachers
        (ID_No, Name, Domain, Department)
 VALUES
        (1345, "Rani", "C Programming", "CSE"),
  (2134, "Murale", "Engineering Graphics", "Civil"),
  (1167, "Geetha", "Computer Architecture", "CSE"),
  (2156, "Vimala", "Electricals", "EEE"),
  (2112, "Ramesh", "Java", "CSE"),
  (1545, "Pranesh", "Mechanics", "Mechanical"),
  (1790, "Krishna", "Structures", "Civil"),
  (1645, "Radhika", "Data Communications", "ECE"),
  (2345, "Santha", "Object Oriented Design", "CSE"),
  (1851, "ManiKumar", "Embedded Systems", "CSE")
```

SQLite

```
1 SELECT * FROM Teachers;
```

ID_No	Name	Domain	Department
1167	Geetha	Computer Architecture	CSE
1345	Rani	C Programming	CSE
1545	Pranesh	Mechanics	Mechanical
1645	Radhika	Data Communications	ECE
1790	Krishna	Structures	Civil
1851	ManiKumar	Embedded Systems	CSE
2112	Ramesh	Java	CSE
2134	Murale	Engineering Graphics	Civil
2156	Vimala	Electricals	EEE
2345	Santha	Object Oriented Design	CSE

4) Fetching all the students from 2020 batch:

SELECT * FROM Students WHERE year_of_batch=2020;

SQLite

```
SELECT * FROM Students WHERE year_of_batch=2020;
```

Registration_No	Name	Age	Mobile_No	Year_of_Batch
2004041	Kala	20	9887956402	2020
2007035	Janani	19	8834512390	2020
2007056	Yalini	20	9034678295	2020
2007078	Ezhil	21	7789345609	2020

5) Fetching all teachers from CS department:

SELECT * FROM Teachers WHERE Department='CSE';

SQLite

```
SELECT * FROM Teachers WHERE Department='CSE';
```

ID_No	Name	Domain	Department
1167	Geetha	Computer Architecture	CSE
1345	Rani	C Programming	CSE
1851	ManiKumar	Embedded Systems	CSE
2112	Ramesh	Java	CSE
2345	Santha	Object Oriented Design	CSE

6) Editing at least 3 records of students

Updating Student age using Primary Key (Registration_No)

```
UPDATE Students
SET Age=20
WHERE registration_no=1903023;
```

Updating Student Name using Primary Key (Registration_No)

```
UPDATE Students
SET Name="Tamizh"
WHERE registration_no=1905031;
```

Updating Student Name using Mobile_No (Unique Column)

```
UPDATE Students
SET Name="Vinay"
WHERE mobile_no=7734978020;
```

Displaying the table:

SQLite

```
SELECT * FROM Students;
```

Registration_No	Name	Age	Mobile_No	Year_of_Batch
1802001	Malathi	21	7845678670	2018
1807001	Aditya	21	9998978670	2018
1901001	Vinay	20	7734978020	2019
1903023	Kamali	20	7834895674	2019
1905031	Tamizh	21	7869023456	2018
2004041	Kala	20	9887956402	2020
2007035	Janani	19	8834512390	2020
2007056	Yalini	20	9034678295	2020
2007078	Ezhil	21	7789345609	2020
2107001	Anaya	19	8968978670	2021

7) Deleting 2 records from teachers table:

Updating Teacher's Department using Domain

```
UPDATE Teachers
SET Department="CSE"
WHERE Domain="Data Communications";
```

Updating Teacher's Name using Primary Key (ID_No)

```
UPDATE Teachers
SET Name="Pranesh Raj"
WHERE ID_No=1545;
```

Displaying the table:

SQLite

```
SELECT * FROM Teachers;
```

ID_No	Name	Domain	Department
1167	Geetha	Computer Architecture	CSE
1345	Rani	C Programming	CSE
1545	Pranesh Raj	Mechanics	Mechanical
1645	Radhika	Data Communications	CSE
1790	Krishna	Structures	Civil
1851	ManiKumar	Embedded Systems	CSE
2112	Ramesh	Java	CSE
2134	Murale	Engineering Graphics	Civil
2156	Vimala	Electricals	EEE
2345	Santha	Object Oriented Design	CSE

ASSIGNMENT 2

1. Create an Interest calculator for banks using java which incorporates, Inheritance, polymorphism, classes, object etc.

2. User can first select a bank.

3. After selecting bank user can select a type of loans like, personal loan, housing loan, educational loan, gold loan.

4. User should be able to enter amount of loan they need. (If gold how many grams)

5. System should be able to present the interest rate along with period of repayment

LoanTest.java

```
package loantest;

import static java.lang.Math.pow;
import java.util.Scanner;

//Base class
class Bank
{
    int loanType;
    double loanAmt;
    double EMI;
    int tenure;

    void emiCalculation(double p, double r)
    {
      //Assuming tenure (in years) based on the loan amount
      if(p < 100000)
          tenure = 1;
      else if(p < 1000000)
          tenure = 5;
      else if(p < 7000000)
          tenure = 7;
      else if(p < 10000000)
          tenure = 10;
      else
          tenure = 15;

      double R = r / (12 * 100); // one month interest
      double T = tenure * 12; // tenure in months
      EMI = (p * R * pow(1 + R, T)) / (pow(1 + R, T) - 1);  //monthly interest
    }

    //printLoanDetails Function with 2 parameters
    void printLoanDetails(double loanAmt, double interestRate)
    {
```

```
        System.out.println("\n========== LOAN DETAILS ========== ");
        System.out.println("LOAN AMOUNT: " + loanAmt);
        System.out.println("RATE OF INTEREST: " + interestRate);
        System.out.println("MONTHLY PAYMENT: " + EMI);
        System.out.println("PERIOD OF REPAYMENT: " + tenure + " years");
    }

    //printLoanDetails Function with 3 parameters
    void printLoanDetails(double loanAmt, double interestRate, int grams)
    {
        System.out.println("\n========== LOAN DETAILS ========== ");
        System.out.println("WEIGHTOF GOLD PLEDGED: " + grams + " grams");
        System.out.println("LOAN AMOUNT: " + loanAmt);
        System.out.println("RATE OF INTEREST: " + interestRate);
        System.out.println("MONTHLY PAYMENT: " + EMI);
        System.out.println("PERIOD OF REPAYMENT: " + tenure + " years");
    }
}

class IndianBank extends Bank
{
    //rate of interest for different types of loans in Indian Bank
    double interestRate_Personal = 7.5;
    double interestRate_Housing = 8;
    double interestRate_Gold = 8.25;
    double interestRate_Car = 6;
    double interestRate_Educational = 6.3;

    //This function assigns the rate of interest based on loan type
    void CalculateLoan(int loanType, double loanAmt, int grams)
    {
        switch (loanType)
        {
            case 1:
                emiCalculation(loanAmt, interestRate_Personal);
                printLoanDetails(loanAmt, interestRate_Personal);
                break;
            case 2:
                emiCalculation(loanAmt, interestRate_Housing);
                printLoanDetails(loanAmt, interestRate_Housing);
                break;
            case 3:
                emiCalculation(loanAmt, interestRate_Gold);
                printLoanDetails(loanAmt, interestRate_Gold, grams);
                break;
            case 4:
                emiCalculation(loanAmt, interestRate_Car);
                printLoanDetails(loanAmt, interestRate_Car);
                break;
            case 5:
                emiCalculation(loanAmt, interestRate_Educational);
                printLoanDetails(loanAmt, interestRate_Educational);
```

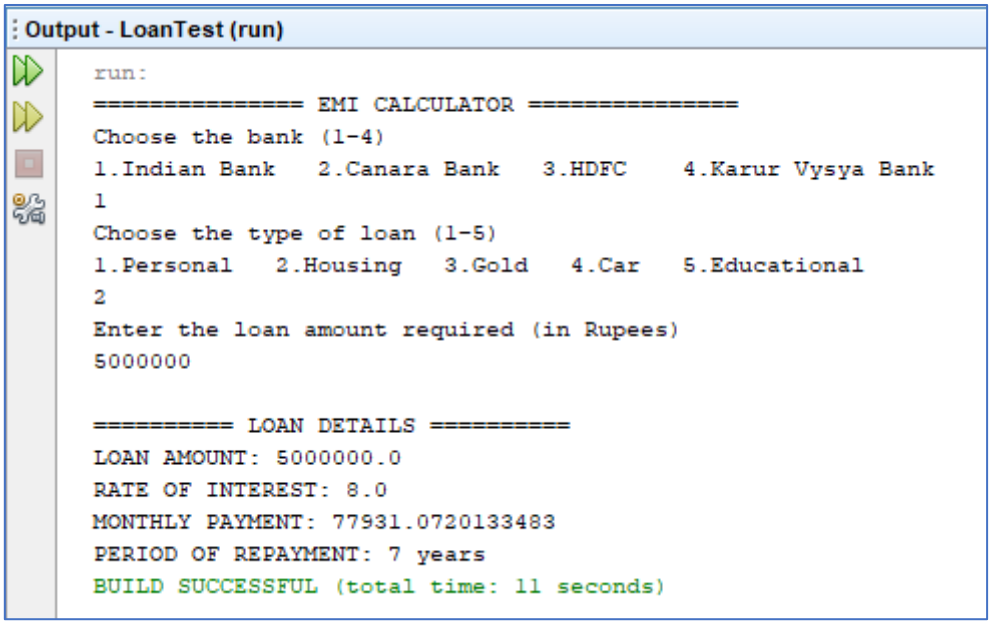
```
            break;
        default:
            break;
      }
   }
}

class CanaraBank extends Bank
{
   //rate of interest for different types of loans in Canara Bank
   double interestRate_Personal = 8.5;
   double interestRate_Housing = 7;
   double interestRate_Gold = 8.25;
   double interestRate_Car = 7;
   double interestRate_Educational = 8;

   //This function assigns the rate of interest based on loan type
   void CalculateLoan(int loanType, double loanAmt, int grams)
   {
      switch (loanType)
      {
         case 1:
            emiCalculation(loanAmt, interestRate_Personal);
            printLoanDetails(loanAmt, interestRate_Personal);
            break;
         case 2:
            emiCalculation(loanAmt, interestRate_Housing);
            printLoanDetails(loanAmt, interestRate_Housing);
            break;
         case 3:
            emiCalculation(loanAmt, interestRate_Gold);
            printLoanDetails(loanAmt, interestRate_Gold, grams);
            break;
         case 4:
            emiCalculation(loanAmt, interestRate_Car);
            printLoanDetails(loanAmt, interestRate_Car);
            break;
         case 5:
            emiCalculation(loanAmt, interestRate_Educational);
            printLoanDetails(loanAmt, interestRate_Educational);
            break;
         default:
            break;
      }
   }
}

class HDFCBank extends Bank
{
   //rate of interest for different types of loans in HDFC Bank
   double interestRate_Personal = 8;
   double interestRate_Housing = 7;
```

```
    double interestRate_Gold = 9;
    double interestRate_Car = 7.5;
    double interestRate_Educational = 7.5;

    //This function assigns the rate of interest based on loan type
    void CalculateLoan(int loanType, double loanAmt, int grams)
    {
       switch (loanType)
       {
          case 1:
             emiCalculation(loanAmt, interestRate_Personal);
             printLoanDetails(loanAmt, interestRate_Personal);
             break;
          case 2:
             emiCalculation(loanAmt, interestRate_Housing);
             printLoanDetails(loanAmt, interestRate_Housing);
             break;
          case 3:
             emiCalculation(loanAmt, interestRate_Gold);
             printLoanDetails(loanAmt, interestRate_Gold, grams);
             break;
          case 4:
             emiCalculation(loanAmt, interestRate_Car);
             printLoanDetails(loanAmt, interestRate_Car);
             break;
          case 5:
             emiCalculation(loanAmt, interestRate_Educational);
             printLoanDetails(loanAmt, interestRate_Educational);
             break;
          default:
             break;
       }
    }
}

class KVB_Bank extends Bank
{
    //rate of interest for different types of loans in Karur Vysya Bank
    double interestRate_Personal = 6.5;
    double interestRate_Housing = 7.6;
    double interestRate_Gold = 8;
    double interestRate_Car = 5;
    double interestRate_Educational = 9.5;

    //This function assigns the rate of interest based on loan type
    void CalculateLoan(int loanType, double loanAmt, int grams)
    {
       switch (loanType)
       {
          case 1:
             emiCalculation(loanAmt, interestRate_Personal);
             printLoanDetails(loanAmt, interestRate_Personal);
```

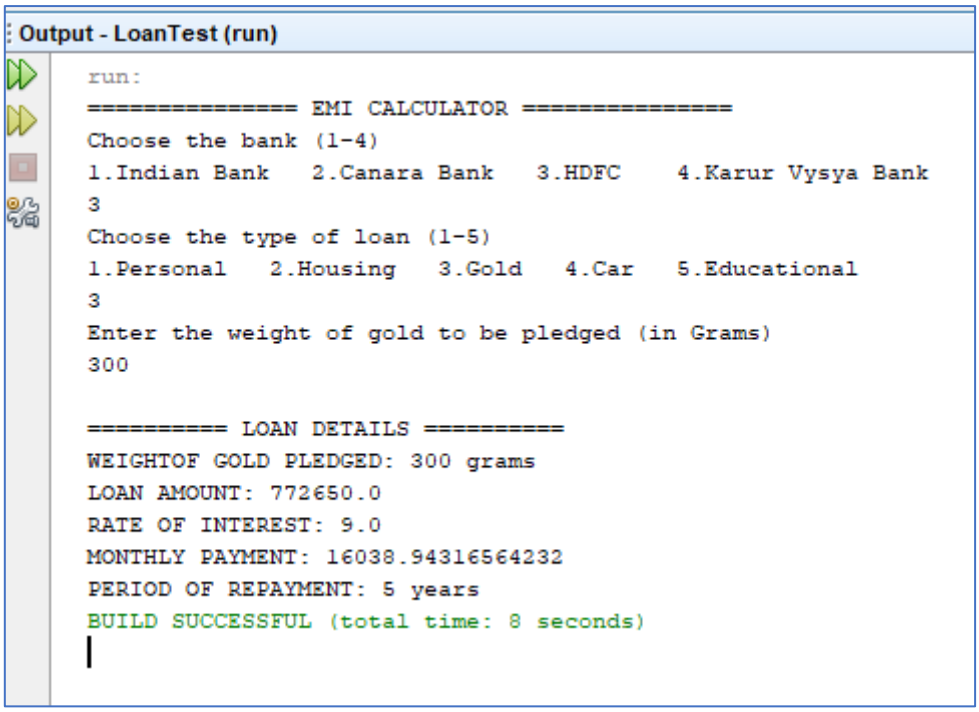
```
            break;
        case 2:
            emiCalculation(loanAmt, interestRate_Housing);
            printLoanDetails(loanAmt, interestRate_Housing);
            break;
        case 3:
            emiCalculation(loanAmt, interestRate_Gold);
            printLoanDetails(loanAmt, interestRate_Gold, grams);
            break;
        case 4:
            emiCalculation(loanAmt, interestRate_Car);
            printLoanDetails(loanAmt, interestRate_Car);
            break;
        case 5:
            emiCalculation(loanAmt, interestRate_Educational);
            printLoanDetails(loanAmt, interestRate_Educational);
            break;
        default:
            break;
      }
   }
}

public class LoanTest {

   public static void main(String[] args)
   {
      Scanner s = new Scanner(System.in);
      int bankNo;
      int loanType;
      double loanAmt;
      int grams=0;
      System.out.println("================ EMI CALCULATOR
================ ");
      System.out.println("Choose the bank (1-4)");
      System.out.println("1.Indian Bank   2.Canara Bank   3.HDFC    4.Karur
Vysya Bank");
      bankNo = s.nextInt();
      System.out.println("Choose the type of loan (1-5)");
      System.out.println("1.Personal   2.Housing   3.Gold   4.Car
5.Educational");
      loanType = s.nextInt();
      if(loanType != 3)
      {
         System.out.println("Enter the loan amount required (in Rupees)");
         loanAmt = s.nextDouble();
      }
      else     //in case of gold loan
      {
         System.out.println("Enter the weight of gold to be pledged (in Grams)");
         grams = s.nextInt();
         //Loan sanctioned for 75% of the amount gold's worth
```

```
            //Assumed that 1 gram of gold is worth Rs.3434/-
            loanAmt = grams*3434*0.75;
        }

        //Creates object for the corresponding bank's class and calls the necessary
function
        switch(bankNo)
        {
            case 1:
                IndianBank obj_indianBank = new IndianBank();
                obj_indianBank.CalculateLoan(loanType, loanAmt, grams);
                break;
            case 2:
                CanaraBank obj_canaraBank = new CanaraBank();
                obj_canaraBank.CalculateLoan(loanType, loanAmt, grams);
                break;
             case 3:
                HDFCBank obj_HDFCBank = new HDFCBank();
                obj_HDFCBank.CalculateLoan(loanType, loanAmt, grams);
                break;
             case 4:
                KVB_Bank obj_KVB_Bank = new KVB_Bank();
                obj_KVB_Bank.CalculateLoan(loanType, loanAmt, grams);
                break;
             default:
                 System.out.println("Choose among the given banks (1-4)");
                 break;
        }
    }
}
```

OUTPUT 1:



OUTPUT 2:



*Compiled using NetBeans IDE 8.2