Weak\_Four\_Assignment\_Tosif\_Mansuri

package weakfour;

import java.sql.Timestamp;

import java.util.Date;

public class Bill {

private int billId;

private int consumerId;

private String billDate;

private String billDescription;

private double billAmount;

//create default constructor

public Bill() {}

//create parameter constructor

public Bill(int billId, int consumerId, String strdate, String billDescription, double billAmount) {

super();

this.billId = billId;

this.consumerId = consumerId;

this.billDate = strdate;

this.billDescription = billDescription;

this.billAmount = billAmount;

}

//create getter and setter mathod

public int getBillId() {

return billId;

}

public void setBillId(int billId) {

this.billId = billId;

}

public int getConsumerId() {

return consumerId;

}

public void setConsumerId(int consumerId) {

this.consumerId = consumerId;

}

public String getBillDate() {

return billDate;

}

public void setBillDate(String billDate2) {

this.billDate = billDate2;

}

public String getBillDescription() {

return billDescription;

}

public void setBillDescription(String billDescription) {

this.billDescription = billDescription;

}

public double getBillAmount() {

return billAmount;

}

public void setBillAmount(double billAmount) {

this.billAmount = billAmount;

}

@Override

// create to string mathod

public String toString() {

return "Bill [billId=" + billId + ", consumerId=" + consumerId + ", billDate=" + billDate + ", billDescription="

+ billDescription + ", billAmount=" + billAmount + "]";

}

}

package weakfour;

import java.sql.Timestamp;

import java.util.Date;

import java.util.List;

public class BillOperationImpl implements IBillOperation{

//create Array

Bill[]bill=new Bill[100];

static int index;

//create save data record mathod

@Override

public int saveBillRecord(Bill b) {

bill[index]=b;

index++;

System.out.println("Employee has been Added:");

return 0;

}

// create edit data record mathod

@Override

public int editBillRecord(int billId, int consumerId, String billDate, String billDescription, double billAmount) {

// TODO Auto-generated method stub

boolean edited = false;

for(int i =0;i<index;i++) {

if(bill[i].getBillId()==billId) {

bill[i].setConsumerId(consumerId);

bill[i].setBillDate(billDate);

bill[i].setBillDescription(billDescription);

bill[i].setBillAmount(billAmount);

edited=true;

break;

}

if(edited)

System.out.println("Employee details edited");

else

System.out.println("Employee not found");

}

return 0;

}

//create remove mathod

@Override

public int removeBillRecord(int billId) {

for(int i=0;i<index;i++) {

if(bill[i].getBillId()==billId) {

bill[i].setConsumerId(-1);

bill[i].setBillDate(null);

bill[i].setBillDescription(null);

bill[i].setBillAmount(-1);

}

else {

System.out.println("Bill id not found");

}

}

return billId;

}

// create get all record

@Override

public List<Bill> getAllBillRecord() {

for(int i=0;i<index;i++)

{

System.out.println(bill[i]);

}

return null;

}

// create gettBill record mathod to by id

@Override

public Bill getBillRecordById(int billl) {

for (int i=0;i<index;i++) {

if(bill[i].getBillId()==billl) {

System.out.println(bill[i]);

}

else

System.out.println("Employee id not found");

}

return null;

}

}

package weakfour;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.sql.Timestamp;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.Scanner;

public class BillOperationMain {

public static void main(String[] args) throws ParseException {

// create object

BillOperationImpl billImpl=new BillOperationImpl();

Scanner sc=new Scanner(System.in);

//create do while loop

do {

try {

Connection con = null;

try {

con = DBConnection.getConnection();

}

catch(Exception e) {

e.printStackTrace();

}

//Write and Execute query

Statement st=con.createStatement();

int billId;

int consumerId;

// create date formate

java.sql.Date sqldate;

java.util.Date utildate;

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

String strdate="1-3-2021";

utildate=sdf.parse(strdate);

sqldate = new java.sql.Date(utildate.getTime());

String billDescription;

double billAmount;

ResultSet a;

int ch;

System.out.println("Select the operation to perform:");

System.out.println("1.Save \n2.Edit\n3.Delete\n4.FetchAll \n5.FetchById");

System.out.println("enter your choice:");

ch=sc.nextInt();

//create switch statement

switch(ch) {

case 1:

System.out.println("Enter BillId id: ");

billId=sc.nextInt();

System.out.println("Enter Consumer id : ");

consumerId=sc.nextInt();

System.out.println("Enter bill Date : ");

sc.nextLine();

strdate=sc.nextLine();

System.out.println("Enter Bill Description: ");

sc.nextLine();

billDescription=sc.nextLine();

System.out.println("Enter Bill amount: ");

billAmount=sc.nextDouble();

String sql2="insert into bill values("+billId+","+consumerId+",'"+strdate+"','"+billDescription+"',"+billAmount+")";

Bill b1=new Bill(billId,consumerId,strdate,billDescription,billAmount);

billImpl.saveBillRecord(b1);

ch=st.executeUpdate(sql2);

System.out.println("\n \*\_\_Inserted\_\_\* \n");

break;

case 2:// Edit

System.out.println("Enter the Employee id which u want to edit:");

System.out.println("Enter Bill id: ");

billId=sc.nextInt();

System.out.println("Enter Consumer id : ");

consumerId=sc.nextInt();

System.out.println("Enter Bill Date : ");

sc.nextLine();

strdate=sc.nextLine();

System.out.println("Enter Bill Description: ");

sc.nextLine();

billDescription=sc.nextLine();

System.out.println("Enter Bill amount: ");

billAmount=sc.nextDouble();

String sql3="update bill set billId=("+billId+"),consumerId=("+consumerId+"),billDate=('"+strdate+"'),billDescription=('"+billDescription+"'),billAmount=("+billAmount+") where billId=("+billId+")";

billImpl.editBillRecord(billId, consumerId, strdate, billDescription, billAmount);

ch= st.executeUpdate(sql3);

System.out.println("...Edited...");

break;

case 3: //delete

System.out.println("Enter id number : ");

billId=sc.nextInt();

String sql1="delete from bill where billid=("+billId+")";

billImpl.removeBillRecord(billId);

ch=st.executeUpdate(sql1);

System.out.println("\n \*\_\_Delete succesfull\_\_\* \n");

break;

case 4://get all record for a databases

billImpl.getAllBillRecord();

String sqlq="select \* from bill";

ResultSet rs=st.executeQuery(sqlq);

while(rs.next())

{

System.out.println(rs.getInt(1)+" "+rs.getInt(2)+" "+rs.getDate(3)+" "+rs.getString(4)+" "+rs.getDouble(5));

}

break;

case 5: //get fetch record for specific id

System.out.println("Enter BillId number : ");

billId=sc.nextInt();

billImpl.getBillRecordById(billId);

String sql4="Select \* from bill where billId=("+billId+")";

a = st.executeQuery(sql4);

while(a.next()) {

System.out.println(a.getInt(1)+" "+a.getInt(2)+" "+a.getDate(3)+" "+a.getString(4)+" "+a.getDouble(5)); }

break;

}

}

catch(SQLException e1)

{

System.out.println(e1.getMessage());

}

}while(true);

}

}

package weakfour;

import java.sql.Timestamp;

import java.util.Date;

import java.util.List;

//create interface

public interface IBillOperation {

// create all mathods

int saveBillRecord(Bill b);

int editBillRecord(int billId,int consumerId,String billDate,String billDescription,double billAmount);

int removeBillRecord(int bill);

List<Bill> getAllBillRecord();

Bill getBillRecordById(int bill);

}

package weakfour;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.Scanner;

public class DBConnection {

public static Connection getConnection() throws ClassNotFoundException, SQLException

{

//cretae server root and password

String driver="com.mysql.cj.jdbc.Driver";

String dburl="jdbc:mysql://localhost:3306/billpayment";

String user="root";

String password="root";

// load the driver

Class.forName(driver);

//create the connection

Connection con = DriverManager.getConnection(dburl,user,password);

return con;

}

}

package weakfour;

import static org.junit.jupiter.api.Assertions.\*;

import org.junit.jupiter.api.Test;

// create test class for BillOperation Test casses

class BillOperationTest {

BillOperationImpl b = new BillOperationImpl();

@Test

void testSaveBillRecord() {

assertEquals(0,b.getBill());

}

@Test

void testEditBillRecord() {

assertEquals(0,b.editBillRecord(0, 0, null, null, 0));

}

@Test

void testRemoveBillRecord() {

assertEquals(0,b.removeBillRecord(0));

}

@Test

void testGetAllBillRecord() {

assertEquals(null,b.getAllBillRecord());

}

@Test

void testGetBillRecordById() {

assertEquals(null,b.getBillRecordById(0));

}

}

package weakfour;

import static org.junit.jupiter.api.Assertions.\*;

import org.junit.jupiter.api.Test;

//create test cases for bill class

class BillTest {

Bill b=new Bill();

@Test

void testGetBill() {

assertEquals(0,b.getBillId());

}

@Test

void testGetConsumerld() {

assertEquals(0,b.getConsumerId());

}

@Test

void testGetBilldate() {

assertEquals(null,b.getBillDate());

}

@Test

void testGetBillDescription() {

assertEquals(null,b.getBillDescription());

}

@Test

void testGetBillamount() {

assertEquals(0,b.getBillAmount());

}

}