```
password
```

```
postgres

```

```java
/**
 * @author david benzaquen
 */
public class Sqlmaker {
    /**
     * his function turns the parameters to a insert into student using sql
     * @param lname student last name
     * @param fname student first name
     * @param standing  student standing
     * @param credits amount of credits
     * @param amount_owed amount the student owed
     * @return a string of the sql quarry
     */
    public String insertStudent(String lname, String fname, String standing, int credits, float amount_owed) {
        StringBuilder sql = new StringBuilder("INSERT INTO STUDENTS ( lname, fname, standing, gpa, credits, amount_owed) values ( ");
        sql.append(lname);
        sql.append(", ");
        sql.append(fname);
        sql.append(", ");
        sql.append(standing);
        sql.append(", ");
        sql.append(credits);
        sql.append(", ");
        sql.append(amount_owed);
        sql.append(");");


        return sql.toString();
    }

    /**
     * this function turns the parameters to a insert into student using sql
     * @param lname students last name
     * @param fname students first name
     * @param standing students standing
     * @param credits amount of credits
     * @return a string of the sql quarry
     */
    public String insertStudent (String lname, String fname, String standing, int credits) {
        StringBuilder sql = new StringBuilder("INSERT INTO STUDENTS ( lname, fname, standing, gpa, credits) values ( ");
        sql.append(lname);
        sql.append(", ");
        sql.append(fname);
        sql.append(", ");
        sql.append(standing);
        sql.append(", ");
        sql.append(credits);
        sql.append(");");

        return sql.toString();
    }

    /**
```

```java
     * this function creates the sql quarry for insert into teacher
     * @param lname teachers last name
     * @param fname teachers first name
     * @param salary teachers salary
     * @param dept teachers dept
     * @return a string of the quarry
     */
    public String inserteacher(String lname, String fname, float salary, String dept) {
        StringBuilder sql = new StringBuilder( "INSERT INTO TEACHERS (lname, fname, salary, dept) values (");

        sql.append(lname);
        sql.append(", ");
        sql.append(fname);
        sql.append(", ");
        sql.append(salary);
        sql.append(", ");
        sql.append(dept);
        sql.append(");");

        return sql.toString();

    }

    /**
     * this function creates the sql quarry for insert into teacher
     * @param lname teachers last name
     * @param fname teachers first name
     * @param salary teachers salary
     * @return a string of the quarry
     */
    public String inserteacher(String lname, String fname, float salary ) {
        StringBuilder sql = new StringBuilder("INSERT INTO TEACHERS ( lname, fname, salary, dept) values (");

        sql.append(lname);
        sql.append(", ");
        sql.append(fname);
        sql.append(", ");
        sql.append(salary);
        sql.append(");");
        return sql.toString();

    }

    /**
     * this functions the string for the insert into classes  quarry
     * @param amount_enrolled this is the amount of students enrolled into the class
     * @param teacherid this is the teacher id number
     * @param emplid this the the emplid of the
     * @return a string of the quarry
     */
    public  String insertClasses(int amount_enrolled, int teacherid, int emplid ){
        StringBuilder sql = new StringBuilder("insert into classes ( amount_enrolled, teacherid ,emplid) values (");
        sql.append(amount_enrolled);
        sql.append(", ");
        sql.append(teacherid);
        sql.append(", ");
        sql.append(emplid);
        sql.append(");");
        return sql.toString();

    }

    /**
     * creates a quarry to find a students information
     * @param emplid student emplid
     * @return the string of the quarry
     */
    public String findStudnetInfo( int emplid) {

        StringBuilder sql = new StringBuilder("SELECT * FROM students WHERE emplid =");
        sql.append(emplid);
        sql.append(";");

        return sql.toString();
    }

    /**
     * this method creates a quarry to find all information for a teacher with the given teacherid
     * @param teacherid teacher id
     * @return the quarry in the form of a string
     */
    public  String findTeacherInfo(int teacherid){
        StringBuilder sql = new StringBuilder("SELECT * from teachers WHERE teachers = ");
        sql.append(teacherid);
        sql.append(";");

        return sql.toString();
    }

    /**
     * this method creates a quarry to find all information for a teacher with the given teacherid
     * @param classid class id
     * @return the quarry in the form of a string
     */
    public  String findClassInformation(int classid){
        StringBuilder sql = new StringBuilder("SELECT * from classes WHERE classid =");
        sql.append(classid);
        sql.append(";");
        return sql.toString();

    }

    /**
     *   creates the quarry to adds amount to amount owed
     * @param emplid students emplid number
     * @param amount_owed amount to be added
     * @return string of the sql quarry
     */
    public String addToAmountOwed(int emplid, float amount_owed){
        StringBuilder sql = new StringBuilder("UPDATE students SET amount_owed= amount_owed + ");
        sql.append(amount_owed);
        sql.append("WHERE emplid=");
        sql.append(emplid);
        sql.append(";");
        return sql.toString();

    }

    /**
     *creates the quarry to adds amount to amount owed
     * @param emplid student emplid number
     * @param amount_owed amount to be subtracted
     * @return string of the sql quarry
     */
    public String subtractAmountOwed (int emplid, float amount_owed){
        StringBuilder sql = new StringBuilder("UPDATE students SET amount_owed= amount_owed - ");
        sql.append(amount_owed);
        sql.append("WHERE emplid=");
        sql.append(emplid);
        sql.append(";");
        return sql.toString();

    }





}
```

```java




import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

public class SQLDriver {

    public static void main(String[] args) {
        //implements endl in java
        final String endl = System.getProperty("line.separator");

        //creates an scanner object to get user imput
        Scanner input = new Scanner(System.in);

        SQLConnector db1 = new SQLConnector();

        Sqlmaker sqlmaker1 = new Sqlmaker();

        while(true){
            System.out.println("press one to insert press two to make a quarry");
             int flag=input.nextInt();
             restart:
            switch (flag){
                case 1 :
                {
                    int x=0;
                    System.out.println("press  1 to inset in to student, 2 to insert in to classes and, 3 to insert into teachers");
                    x=input.nextInt();
                    switch (x){
                        //inset into student
                        case 1 :
                        {



                            System.out.println("please enter lname");
                            String lname=input.nextLine();

                            input.nextLine();
                            System.out.println("please enter fname");
                            String fname = input.nextLine();

                            input.nextLine();
                            System.out.println("please enter standing");
                            String standing = input.nextLine();

                            System.out.println("please enter the amount of credits");
                            int cedits = input.nextInt();

                            System.out.println("please enter the amount owed");
                            float  amount_owed=input.nextFloat();
```

```


                          db1.stmtMaker(sqlmaker1.insertStudent(lname, fname, standing, cedits,amount_owed ),db1.c);


                        }
                        //insert into classes
                        case 2:
                            {
                            System.out.println("please enter the amount enrolled");
                            int amount_enrolled=input.nextInt();


                            System.out.println("please enter the teacher id");
                            int teacherid = input.nextInt();

                            System.out.println("please enter the emplid");
                            int emplid = input.nextInt();

                            db1.stmtMaker(sqlmaker1.insertClasses(amount_enrolled,teacherid,emplid),db1.c);
                        }
                        //insert into teachers
                        case 3:
                        {
                            System.out.println("plese enter the lname ");
                            String lname = input.nextLine();

                            input.nextLine();
                            System.out.println("please enter the fname ");
                            String fname = input.nextLine();

                            input.nextLine();
                            System.out.println("please enter the salary");
                            float salary = input.nextFloat();

                            System.out.println("please enter the dept ");
                            String dept = input.nextLine();

                            db1.stmtMaker(sqlmaker1.inserteacher(lname,fname,salary,dept),db1.c);
                        }
                        default:
                        {
                            System.out.println("error you didnt pick anything");

                        }
                    }

                }
```

```java

                case 2 :
                {
                    System.out.println("press 1 to find student information, press 2 to find teacher information, press 3 to find class  information"+
                            endl+"press 4 to add to students amount owed and press five to subtract to students amount");
                    int x = input.nextInt();


                    switch (x)
                    {//find student informtion
                        case 1: {
                            ResultSet rs = null;
                            System.out.println("please enter the emplid");
                            int emplid = input.nextInt();
                            rs = db1.stmtMaker(sqlmaker1.findStudnetInfo(emplid), db1.c);
                            printStudentInformation(rs);

                        }

                        //find teacher information
                        case 2:
                        {
                            ResultSet rs = null;
                            System.out.println("please enter the teacherid");
                            int teacherid = input.nextInt();
                            rs =db1.stmtMaker(sqlmaker1.findTeacherInfo(teacherid),db1.c);
                            printTeacherInformation(rs);
                        }
                        //find class information
                        case 3:
                        {
                            ResultSet rs = null;
                            System.out.println("please enter the class number");
                            int classid = input.nextInt();
                            rs=db1.stmtMaker(sqlmaker1.findClassInformation(classid),db1.c);
                            printClassInformation(rs);
                        }
                        //add amount owed
                        case 4:
                        {
                            ResultSet rs = null;
                            System.out.println("please enter the emplid");
                            int emplid = input.nextInt();
                            System.out.println("please enter the amount owed");
                            float amount_owed = input.nextFloat();
                            rs = db1.stmtMaker(sqlmaker1.addToAmountOwed(emplid,amount_owed),db1.c);
                            printUpdatedOwed(rs);

                        }
                        //subtact amount owed
                        case 5:
                        {
                            ResultSet rs = null;
                            System.out.println("please enter the emplid");
                            int emplid = input.nextInt();
                            System.out.println("please enter the amount owed");
                            float amount_owed = input.nextFloat();
                            rs = db1.stmtMaker(sqlmaker1.subtractAmountOwed(emplid,amount_owed),db1.c);
                            printUpdatedOwed(rs);
                        }
                        default:
                            System.out.print("error please try again");

                } }

            }
        }









}
    public static void printStudentInformation(ResultSet rs){
            try {


                while (rs.next()){
                    //retiving data from result set
                    int emplid = rs.getInt("emplid");
                    String lname = rs.getString("lname");
                    String fname = rs.getString("fname");
                    String standing = rs.getString("standing");
                    int credits = rs.getInt("credits");
                    float amount_owed = rs.getFloat("amount_owed");

                    //printing results
                    System.out.print("emplid: "+emplid);
                    System.out.print(" lname: "+lname);
                    System.out.print(" fname: "+fname);
                    System.out.print(" standing: "+standing);
                    System.out.print(" credits: "+credits);
                    System.out.print(" amount_owed: "+credits);
                    System.out.println();



                }
            }catch (SQLException e){
                e.printStackTrace();
            }
        }
    public static void printTeacherInformation(ResultSet rs ){
        try {

            while (rs.next()){
                //retirive data from result ret
                String lname =rs.getString("lname");
                String fname = rs. getString("fname");
                float salary = rs.getFloat("salary");
                String dept = rs.getString("dept");
                int teacherid= rs.getInt("teacherid");

                //print results
                System.out.print("teacherid: "+teacherid);
                System.out.print(" lname: "+lname);
                System.out.print(" fname: "+fname);
                System.out.print(" salary: "+salary);
                System.out.print(" dept: "+dept);
                System.out.println();
            }
        }catch (SQLException e){
            e.printStackTrace();
        }
    }

    public static void printClassInformation( ResultSet rs){
        try{
            while (rs.next()){
                //retieve data from result set
                int amount_enrolled = rs.getInt("amount_enrolled");
                int teacherid = rs.getInt("teacherid");
                int emplid = rs.getInt("emplid");
                int classid =rs.getInt("classid");

                //print data from result set
                System.out.print("classid: "+classid);
                System.out.print("teacher: "+teacherid);
                System.out.print("students: "+emplid);
                System.out.print("amount_enrolled: "+amount_enrolled);
                System.out.println();
            }
        }catch (SQLException e){
            e.printStackTrace();
        }
    }
    public static void printUpdatedOwed(ResultSet rs){
        try {
            while (rs.next()){
                int emplid =rs.getInt("emplid");
                float amount_owed =rs.getFloat("amount_owed");

                System.out.print("emplid: "+emplid);
                System.out.print("updated amount owed: "+amount_owed);
                System.out.println();
            }
        }catch(SQLException e){
            e.printStackTrace();
        }
    }

    }
```

```java
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;

public class usercreds {
    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    private String password;
    private String username;
    usercreds(){
        try {
            FileReader userReader = new FileReader(new File("C:\\Users\\david\\IdeaProjects\\sqlProjectForDB\\src\\username.txt"));
            StringBuilder x=new StringBuilder();
            int i;
            while ((i=userReader.read()) != -1){
                x.append(i);
                setUsername(x.toString());
            }
        }catch (FileNotFoundException e){
            e.printStackTrace();
        }catch (IOException e){
            e.printStackTrace();
        }
        try {
            FileReader passReader = new FileReader(new File("C:\\Users\\david\\IdeaProjects\\sqlProjectForDB\\src\\password.txt"));
            StringBuilder x=new StringBuilder();
            int i;
            while ((i=passReader.read()) != -1){
                x.append(i);
                setPassword(x.toString());
            }
        }catch (FileNotFoundException e){
            e.printStackTrace();
        }catch (IOException e){
            e.printStackTrace();
        }





    }



}

```

```java
import java.sql.*;
import java.sql.SQLException;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;


public class SQLConnector {
    usercreds user1=new usercreds();
    private  String user = user1.getUsername();
    private  String pass = user1.getPassword();
    Connection c = null;
    SQLConnector(){



        try{
            //connecting to postgresql using jbdc
            Class.forName("org.postgresql.Driver");
            c=DriverManager.getConnection("jdbc:postgresql://localhost:5433/DBproject",
                    user, pass);

        }catch (ClassNotFoundException e){
            e.printStackTrace();
        }catch (SQLException e){
            e.printStackTrace();
        }












    }

    /**
     * using the connection imputs a quarry with given code
     * @param sql string with the sql code
     * @param c the Connection to the postgresql server
     * @return the result set of the sql quarry
     */
    public ResultSet stmtMaker(String sql, Connection c){
        Statement stmt = null;
        ResultSet rs = null;
        try {
            stmt = c.createStatement();

          rs= stmt.executeQuery(sql);



        }catch (SQLException e) {
            e.printStackTrace();
        }

        return rs;

    }
}

```