```
1 password
```

```
1 postgres
2
```

```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();
    }

    /**
```

```
54      * this function creates the sql quarry for insert into teacher
55      * @param lname teachers last name
56      * @param fname teachers first name
57      * @param salary teachers salary
58      * @param dept teachers dept
59      * @return a string of the quarry
60      */
61     public String inserteacher(String lname, String fname, float
    salary, String dept) {
62         StringBuilder sql = new StringBuilder( "INSERT INTO TEACHERS
     (lname, fname, salary, dept) values (");
63 
64         sql.append(lname);
65         sql.append(", ");
66         sql.append(fname);
67         sql.append(", ");
68         sql.append(salary);
69         sql.append(", ");
70         sql.append(dept);
71         sql.append(");");
72 
73         return sql.toString();
74 
75     }
76 
77     /**
78      * this function creates the sql quarry for insert into teacher
79      * @param lname teachers last name
80      * @param fname teachers first name
81      * @param salary teachers salary
82      * @return a string of the quarry
83      */
84     public String inserteacher(String lname, String fname, float
    salary ) {
85         StringBuilder sql = new StringBuilder("INSERT INTO TEACHERS (
    lname, fname, salary, dept) values (");
86 
87         sql.append(lname);
88         sql.append(", ");
89         sql.append(fname);
90         sql.append(", ");
91         sql.append(salary);
92         sql.append(");");
93         return sql.toString();
94 
95     }
96 
97     /**
98      * this functions the string for the insert into classes  quarry
99      * @param amount_enrolled this is the amount of students enrolled
     into the class
100      * @param teacherid this is the teacher id number
101      * @param emplid this the the emplid of the
102      * @return a string of the quarry
103      */
104     public  String insertClasses(int amount_enrolled, int teacherid,
    int emplid ){
105         StringBuilder sql = new StringBuilder("insert into classes (
    amount_enrolled, teacherid ,emplid) values (");
```

```
106         sql.append(amount_enrolled);
107         sql.append(", ");
108         sql.append(teacherid);
109         sql.append(", ");
110         sql.append(emplid);
111         sql.append(");");
112         return sql.toString();
113 
114     }
115 
116     /**
117      * creates a quarry to find a students information
118      * @param emplid student emplid
119      * @return the string of the quarry
120      */
121     public String findStudnetInfo( int emplid) {
122 
123         StringBuilder sql = new StringBuilder("SELECT * FROM students
     WHERE emplid =");
124         sql.append(emplid);
125         sql.append(";");
126 
127         return sql.toString();
128     }
129 
130     /**
131      * this method creates a quarry to find all information for a
    teacher with the given teacherid
132      * @param teacherid teacher id
133      * @return the quarry in the form of a string
134      */
135     public  String findTeacherInfo(int teacherid){
136         StringBuilder sql = new StringBuilder("SELECT * from teachers
     WHERE teachers = ");
137         sql.append(teacherid);
138         sql.append(";");
139 
140         return sql.toString();
141     }
142 
143     /**
144      * this method creates a quarry to find all information for a
    teacher with the given teacherid
145      * @param classid class id
146      * @return the quarry in the form of a string
147      */
148     public  String findClassInformation(int classid){
149         StringBuilder sql = new StringBuilder("SELECT * from classes
    WHERE classid =");
150         sql.append(classid);
151         sql.append(";");
152         return sql.toString();
153 
154     }
155 
156     /**
157      *   creates the quarry to adds amount to amount owed
158      * @param emplid students emplid number
159      * @param amount_owed amount to be added
```

```
160      * @return string of the sql quarry
161      */
162     public String addToAmountOwed(int emplid, float amount_owed){
163         StringBuilder sql = new StringBuilder("UPDATE students SET
    amount_owed= amount_owed + ");
164         sql.append(amount_owed);
165         sql.append("WHERE emplid=");
166         sql.append(emplid);
167         sql.append(";");
168         return sql.toString();
169 
170     }
171 
172     /**
173      *creates the quarry to adds amount to amount owed
174      * @param emplid student emplid number
175      * @param amount_owed amount to be subtracted
176      * @return string of the sql quarry
177      */
178     public String subtractAmountOwed (int emplid, float amount_owed){
179         StringBuilder sql = new StringBuilder("UPDATE students SET
    amount_owed= amount_owed - ");
180         sql.append(amount_owed);
181         sql.append("WHERE emplid=");
182         sql.append(emplid);
183         sql.append(";");
184         return sql.toString();
185 
186     }
187 
188 
189 
190 
191 
192 }
```

```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();
```

```
55
56
57                          db1.stmtMaker(sqlmaker1.insertStudent(lname
   , fname, standing, cedits,amount_owed ),db1.c);
58
59
60                      }
61                      //insert into classes
62                      case 2:
63                          {
64                          System.out.println("please enter the
   amount enrolled");
65                          int amount_enrolled=input.nextInt();
66
67
68                          System.out.println("please enter the
   teacher id");
69                          int teacherid = input.nextInt();
70
71                          System.out.println("please enter the
   emplid");
72                          int emplid = input.nextInt();
73
74                          db1.stmtMaker(sqlmaker1.insertClasses(
   amount_enrolled,teacherid,emplid),db1.c);
75                      }
76                      //insert into teachers
77                      case 3:
78                      {
79                          System.out.println("plese enter the lname
   ");
80                          String lname = input.nextLine();
81
82                          input.nextLine();
83                          System.out.println("please enter the
   fname ");
84                          String fname = input.nextLine();
85
86                          input.nextLine();
87                          System.out.println("please enter the
   salary");
88                          float salary = input.nextFloat();
89
90                          System.out.println("please enter the dept
   ");
91                          String dept = input.nextLine();
92
93                          db1.stmtMaker(sqlmaker1.inserteacher(
   lname,fname,salary,dept),db1.c);
94                      }
95                      default:
96                      {
97                          System.out.println("error you didnt pick
   anything");
98
99                      }
100                 }
101
102             }
```

```
103
104                 case 2 :
105                 {
106                     System.out.println("press 1 to find student
    information, press 2 to find teacher information, press 3 to find
    class  information"+
107                             endl+"press 4 to add to students amount
    owed and press five to subtract to students amount");
108                     int x = input.nextInt();
109
110
111                     switch (x)
112                     {//find student informtion
113                         case 1: {
114                             ResultSet rs = null;
115                             System.out.println("please enter the
    emplid");
116                             int emplid = input.nextInt();
117                             rs = db1.stmtMaker(sqlmaker1.
    findStudnetInfo(emplid), db1.c);
118                             printStudentInformation(rs);
119
120                         }
121
122                         //find teacher information
123                         case 2:
124                         {
125                             ResultSet rs = null;
126                             System.out.println("please enter the
    teacherid");
127                             int teacherid = input.nextInt();
128                             rs =db1.stmtMaker(sqlmaker1.
    findTeacherInfo(teacherid),db1.c);
129                             printTeacherInformation(rs);
130                         }
131                         //find class information
132                         case 3:
133                         {
134                             ResultSet rs = null;
135                             System.out.println("please enter the
    class number");
136                             int classid = input.nextInt();
137                             rs=db1.stmtMaker(sqlmaker1.
    findClassInformation(classid),db1.c);
138                             printClassInformation(rs);
139                         }
140                         //add amount owed
141                         case 4:
142                         {
143                             ResultSet rs = null;
144                             System.out.println("please enter the
    emplid");
145                             int emplid = input.nextInt();
146                             System.out.println("please enter the
    amount owed");
147                             float amount_owed = input.nextFloat();
148                             rs = db1.stmtMaker(sqlmaker1.
    addToAmountOwed(emplid,amount_owed),db1.c);
149                             printUpdatedOwed(rs);
```

```
150
151                         }
152                         //subtact amount owed
153                         case 5:
154                         {
155                             ResultSet rs = null;
156                             System.out.println("please enter the
    emplid");
157                             int emplid = input.nextInt();
158                             System.out.println("please enter the
    amount owed");
159                             float amount_owed = input.nextFloat();
160                             rs = db1.stmtMaker(sqlmaker1.
    subtractAmountOwed(emplid,amount_owed),db1.c);
161                             printUpdatedOwed(rs);
162                         }
163                         default:
164                             System.out.print("error please try again"
    );
165
166                 } }
167
168             }
169         }
170
171
172
173
174
175
176
177
178 }
179     public static void printStudentInformation(ResultSet rs){
180             try {
181
182
183                 while (rs.next()){
184                     //retiving data from result set
185                     int emplid = rs.getInt("emplid");
186                     String lname = rs.getString("lname");
187                     String fname = rs.getString("fname");
188                     String standing = rs.getString("standing");
189                     int credits = rs.getInt("credits");
190                     float amount_owed = rs.getFloat("amount_owed");
191
192                     //printing results
193                     System.out.print("emplid: "+emplid);
194                     System.out.print(" lname: "+lname);
195                     System.out.print(" fname: "+fname);
196                     System.out.print(" standing: "+standing);
197                     System.out.print(" credits: "+credits);
198                     System.out.print(" amount_owed: "+credits);
199                     System.out.println();
200
201
202
203                 }
204             }catch (SQLException e){
```

```
205                 e.printStackTrace();
206             }
207         }
208     public static void printTeacherInformation(ResultSet rs ){
209         try {
210 
211             while (rs.next()){
212                 //retirive data from result ret
213                 String lname =rs.getString("lname");
214                 String fname = rs. getString("fname");
215                 float salary = rs.getFloat("salary");
216                 String dept = rs.getString("dept");
217                 int teacherid= rs.getInt("teacherid");
218 
219                 //print results
220                 System.out.print("teacherid: "+teacherid);
221                 System.out.print(" lname: "+lname);
222                 System.out.print(" fname: "+fname);
223                 System.out.print(" salary: "+salary);
224                 System.out.print(" dept: "+dept);
225                 System.out.println();
226             }
227         }catch (SQLException e){
228             e.printStackTrace();
229         }
230     }
231 
232     public static void printClassInformation( ResultSet rs){
233         try{
234             while (rs.next()){
235                 //retieve data from result set
236                 int amount_enrolled = rs.getInt("amount_enrolled");
237                 int teacherid = rs.getInt("teacherid");
238                 int emplid = rs.getInt("emplid");
239                 int classid =rs.getInt("classid");
240 
241                 //print data from result set
242                 System.out.print("classid: "+classid);
243                 System.out.print("teacher: "+teacherid);
244                 System.out.print("students: "+emplid);
245                 System.out.print("amount_enrolled: "+amount_enrolled)
    ;
246                 System.out.println();
247             }
248         }catch (SQLException e){
249             e.printStackTrace();
250         }
251     }
252     public static void printUpdatedOwed(ResultSet rs){
253         try {
254             while (rs.next()){
255                 int emplid =rs.getInt("emplid");
256                 float amount_owed =rs.getFloat("amount_owed");
257 
258                 System.out.print("emplid: "+emplid);
259                 System.out.print("updated amount owed: "+amount_owed)
    ;
260                 System.out.println();
261             }
```

```
262         }catch(SQLException e){
263             e.printStackTrace();
264         }
265     }
266
267     }
268
269
270
271
272
273
274
```

```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();
        }




    }
```

```
58
59
60
61 }
62
```

```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) {
```

```
59              e.printStackTrace();
60          }
61
62          return rs;
63
64      }
65  }
66
```