

**JAVA AWT BASED- Online MOOC's year wise student
database management system - SQL CONNECTIVITY
USING JDBC**

A

Report

*Submitted in partial fulfilment of the
Requirements for the award of the Degree of*

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

By

B.AJAYBABU <1602-18-737-062>



Department of Information Technology

Vasavi College of Engineering (Autonomous)

Ibrahimbagh, Hyderabad-31

2020

BONAFIDE CERTIFICATE

*This to certify that the project report titled
“TECHNICAL SKILLS DATA RETRIEVAL SYSTEM”
project work of Mr.B.AJAYBABU bearing Roll.no:1602-
18-737-062 who carried out this project under my
supervision in the IV semester for the academic year
2019-2020.*

Signature
external examiner

B.LEELAVATHY

Signature
internal examiner

ABSTRACT:

Data retrieval means obtaining data from a database management system such as ODBMS. In this case, it is considered that data is represented in a structured way, and there is no ambiguity in data.

In order to retrieve the desired data the user present a set of criteria by a query. Then the DATA BASE MANAGEMENT SYSTEM (DBMS), software for managing databases, selects the demanded data from the database. The retrieved data may be stored in a file, printed, or viewed on the screen.

A query language, such as STRUCTURED QUERY LANGUAGE (SQL), is used to prepare the queries. SQL is an ANSI standardized query language developed specifically to write database queries. Each DBMS may have its own language, but most relational .

Requirement Analysis:

LIST OF TABLES:

☐ TECHNICAL SKILLS

☐ LEARNEDBY

☐ STUDENTS

☐ CERTIFIEDIN

☐ COURSES

LIST OF ATTRIBUTES WITH THEIR DOMAIN TYPES:

1. TECHNICAL SKILLS:

PROGRAMMING_SKILLS VARCHAR₂(20)

WEB VARCHAR₂(20)

GRAPHICAL VARCHAR₂(20)

COMPUTERSKILLS CHAR(20)

TID NUMBER(5)

2. LEARNEDBY:

SINCE DATE

SID VARCHAR₂(25)

TID NUMBER(5)

3. STUDENTS:

SID VARCHAR2(25)

SNAME VARCHAR2(25)

DOB DATE

EMAIL VARCHAR2(256)

YEAR NUMBER(4)

BRANCH VARCHAR2(6)

4.CERTIFIEDIN:

PASSEDON DATE

SID VARCHAR2(25)

CID VARCHAR2(15)

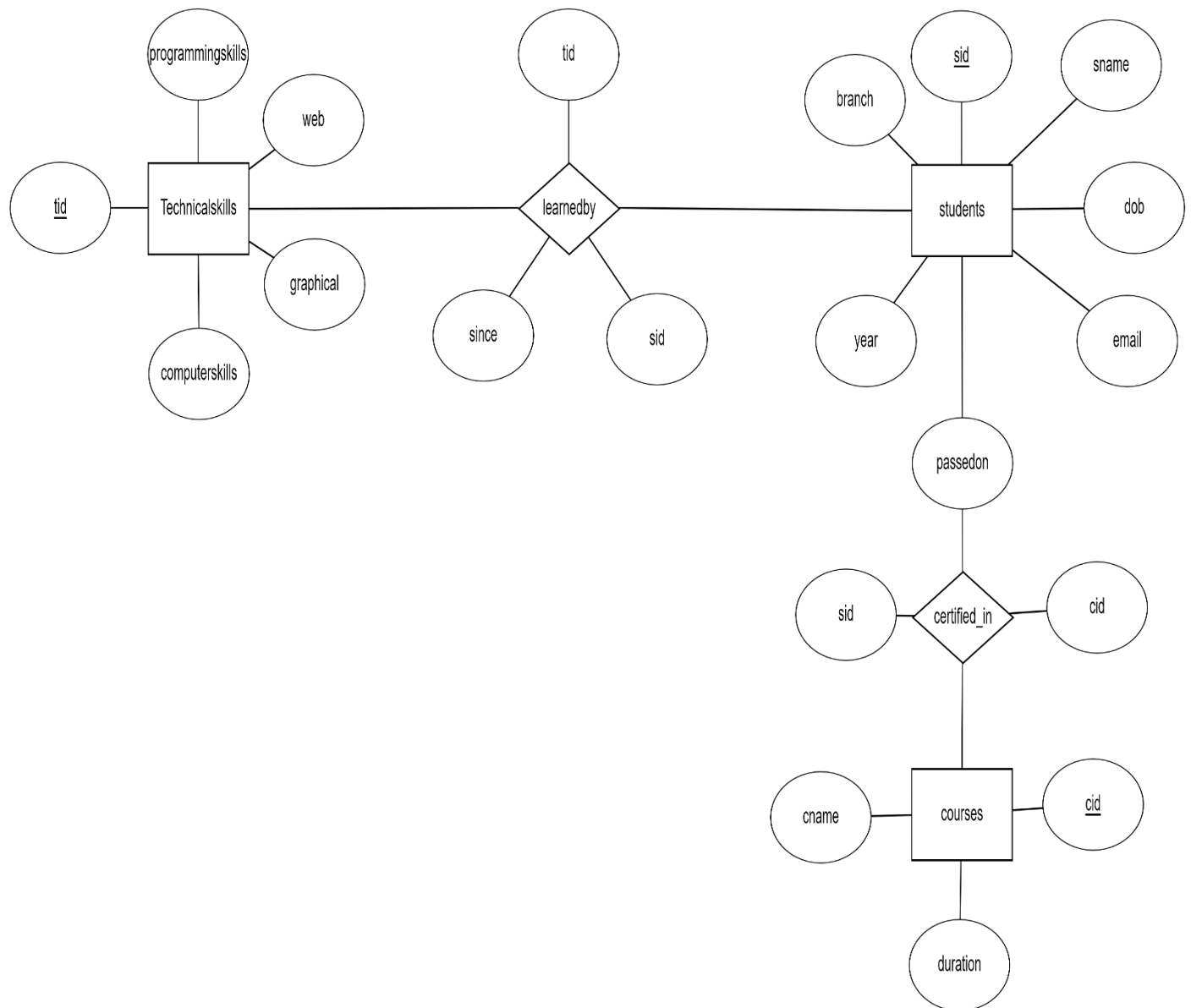
5.COURSES:

CID VARCHAR2(15)

CNAME VARCHAR2(25)

DURATION VARCHAR2(6)

ER DIAGRAM:



MAPPING CARDINALITIES AND PARTICIPATION

CONSTRAINTS:

A Student can learn many courses

ONE-MANY

Technical skills can be learned by any student and any number of

students MANY-ONE

A Student can get certified in any number of courses.

DDL AND DML COMMANDS:

```
SQL> desc technicalskills;
```

Name	Null?	Type
------	-------	------

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

PROGRAMMING_SKILLS		
--------------------	--	--

VARCHAR2(20)		
--------------	--	--

WEB		VARCHAR2(20)
-----	--	--------------

GRAPHICAL		VARCHAR2(20)
-----------	--	--------------

COMPUTERSKILLS		CHAR(20)
----------------	--	----------

TID		NOT NULL NUMBER(5)
-----	--	--------------------

```
SQL> desc students;
```

Name	Null?	Type
------	-------	------

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

SID		NOT NULL VARCHAR2(25)
-----	--	-----------------------

SNAME		VARCHAR2(25)
-------	--	--------------

DOB		DATE
-----	--	------

EMAIL		VARCHAR2(256)
-------	--	---------------

YEAR		NUMBER(4)
------	--	-----------

BRANCH		VARCHAR2(6)
--------	--	-------------

SQL> desc courses;

Name	Null?	Type
------	-------	------

CID	NOT NULL	VARCHAR2(15)
CNAME		VARCHAR2(25)
DURATION		VARCHAR2(6)

SQL> desc learnedby;

Name	Null?	Type
------	-------	------

SINCE		DATE
SID		VARCHAR2(25)
TID		NUMBER(5)

SQL> desc certified_in;

Name	Null?	Type
------	-------	------

PASSEDON

DATE

SID

VARCHAR2(25)

CID

VARCHAR2(15)

Table created.

Table created.

SQL> insert into technicalskills

values('&programming_skills','&web','&graphical','&computerskills');

Enter value for programming_skills: sql

Enter value for web: html

Enter value for graphical: photoshop

Enter value for computerskills: msoffice

old 1: insert into technicalskills

values('&programming_skills','&web','&graphical','&computerskills')

new 1: insert into technicalskills

values('sql','html','photoshop','msoffice')

1 row created.

SQL> /

Enter value for programming_skills: c

Enter value for web: java script

Enter value for graphical: free hand

Enter value for computerskills: powerpoint

old 1: insert into technicalskills

values('&programming_skills','&web','&graphical','&computerskills')

new 1: insert into technicalskills values('c','java script','free

hand','powerpoint')

1 row created.

SQL> /

Enter value for programming_skills: java

Enter value for web: wordpress

Enter value for graphical: acrobat

Enter value for computerskills: spreadsheets

old 1: insert into technicalskills

values('&programming_skills','&web','&graphical','&computerskills')

new 1: insert into technicalskills

values('java','wordpress','acrobat','spreadsheets')

1 row created.

SQL> insert into students

values('&sid','&sname','&dob','&email','&year','&branch');

Enter value for sid: 61

Enter value for sname: Abhiraj.D

Enter value for dob: 14-JAN-2001

Enter value for email: abhiraj13@gmail.com

Enter value for year: 2

Enter value for branch: it

old 1: insert into students

values('&sid','&sname','&dob','&email','&year','&branch')

new 1: insert into students values('61','Abhiraj.D','14-JAN2001','abhiraj13@gmail.com',2,'it')

1 row created.

SQL> /

Enter value for sid: 62

Enter value for sname: ajay.B

Enter value for dob: 13-NOV-2000

Enter value for email: ajaybabu@gmail.com

Enter value for year: 2

Enter value for branch: it

old 1: insert into students

values('&sid','&sname','&dob','&email','&year','&branch')

new 1: insert into students values('62','ajay.B','13-NOV2000','ajaybabu@gmail.com',2,'it')

1 row created.

SQL> /

Enter value for sid: 70

Enter value for sname: sam

Enter value for dob: 01-FEB-2001

Enter value for email: sam007@gmail.com

Enter value for year: 2

Enter value for branch: CSE

old 1: insert into students

values('&sid','&sname','&dob','&email','&year','&branch')

new 1: insert into students values('70','sam','01-FEB2001','sam007@gmail.com',2,'CSE')

1 row created.

SQL> /

Enter value for sid: 311

Enter value for sname: irfan

Enter value for dob: 12-DEC-1999

Enter value for email: irfan@gmail.com

Enter value for year: 3

Enter value for branch: MECH

old 1: insert into students

values('&sid','&sname','&dob','&email','&year','&branch')

new 1: insert into students values('311','irfan','12-DEC1999','irfan@gmail.com',3,'MECH')

1 row created.

SQL> /

Enter value for sid: 100

Enter value for sname: virat

Enter value for dob: 05-NOV-1984

Enter value for email: rcb@gmail.com

Enter value for year: 4

Enter value for branch: Civil

old 1: insert into students

values('&sid','&sname','&dob','&email','&year','&branch')

new 1: insert into students values('100','virat','05-NOV1984','rcb@gmail.com',4,'Civil')

1 row created.

SQL> insert into courses values('&cid','&cname','&duration','&sid');

SQL> /

Enter value for cid: 1

Enter value for cname: java

Enter value for duration: 8weeks

Enter value for sid: 61

old 1: insert into courses values('&cid','&cname','&duration','&sid')

new 1: insert into courses values('1','java','8weeks','61')

1 row created.

SQL> /

Enter value for cid: 2

Enter value for cname: sql

Enter value for duration: 4weeks

Enter value for sid: 62

old 1: insert into courses values('&cid','&cname','&duration','&sid')

new 1: insert into courses values('2','sql','4weeks','62')

1 row created.

SQL> /

Enter value for cid: 3

Enter value for cname: photoshop

Enter value for duration: 8weeks

Enter value for sid: 70

old 1: insert into courses values('&cid','&cname','&duration','&sid')

new 1: insert into courses values('3','photoshop','8weeks','70')

1 row created.

SQL> /

Enter value for cid: 4

Enter value for cname: html

Enter value for duration: 4weeks

Enter value for sid: 311

old 1: insert into courses values('&cid','&cname','&duration','&sid')

new 1: insert into courses values('4','html','4weeks','311')

1 row created.

SQL> /

Enter value for cid: 5

Enter value for cname: spreadsheets

Enter value for duration: 4weeks

Enter value for sid: 100

old 1: insert into courses values('&cid','&cname','&duration','&sid')

new 1: insert into courses values('5','spreadsheets','4weeks','100')

1 row created.

SQL> insert into learnedby values('&since','&sid');

Enter value for since: 1-FEB-2019

Enter value for sid: 61

old 1: insert into learnedby values('&since','&sid')

new 1: insert into learnedby values('1-FEB-2019','61')

1 row created.

SQL> /

Enter value for since: 18-JUN-2019

Enter value for sid: 62

old 1: insert into learnedby values('&since','&sid')

new 1: insert into learnedby values('18-JUN-2019','62')

1 row created.

SQL> /

Enter value for since: 1-JAN-2020

Enter value for sid: 70

old 1: insert into learnedby values('&since','&sid')

new 1: insert into learnedby values('1-JAN-2020','70')

1 row created.

SQL> /

Enter value for since: 10-OCT-2019

Enter value for sid: 311

old 1: insert into learnedby values('&since','&sid')

new 1: insert into learnedby values('10-OCT-2019','311')

1 row created.

SQL> /

Enter value for since: 1-JAN-2020

Enter value for sid: 100

old 1: insert into learnedby values('&since','&sid')

new 1: insert into learnedby values('1-JAN-2020','100')

1 row created.

SQL> insert into certified_in values('&passedon','&sid');

Enter value for passedon: 12-APR-2019

Enter value for sid: 61

old 1: insert into certified_in values('&passedon','&sid')

new 1: insert into certified_in values('12-APR-2019','61')

1 row created.

SQL> /

Enter value for passedon: 19-AUG-2019

Enter value for sid: 62

old 1: insert into certified_in values('&passedon','&sid')

new 1: insert into certified_in values('19-AUG-2019','62')

1 row created.

SQL> /

Enter value for passedon: 01-MAR-2020

Enter value for sid: 70

old 1: insert into certified_in values('&passedon','&sid')

new 1: insert into certified_in values('01-MAR-2020','70')

1 row created.

SQL> /

Enter value for passedon: 30-NOV-2019

Enter value for sid: 311

old 1: insert into certified_in values('&passedon','&sid')

new 1: insert into certified_in values('30-NOV-2019','311')

1 row created.

SQL> /

Enter value for passedon: 13-FEB-2020

Enter value for sid: 100

old 1: insert into certified_in values('&passedon','&sid')

new 1: insert into certified_in values('13-FEB-2020','100')

1 row created.

SQL> select *from courses;

CID CNAME DURATI SID

1 java 8weeks 61

2 sql 4weeks 62

3 photoshop 8weeks 70

4 html 4weeks 311

5 spreadsheets 4weeks 100

SQL> select *from students;

SID SNAME DOB

EMAIL

YEAR BRANCH

61 Abhiraj.D 14-JAN-01

abhiraj13@gmail.com

2 it

62 ajay.B 13-NOV-00

ajaybabu@gmail.com

2 it

SID SNAME DOB

EMAIL

YEAR BRANCH

70 sam 01-FEB-01

sam007@gmail.com

2 CSE

311 irfan 12-DEC-99

irfan@gmail.com

SID SNAME DOB

EMAIL

YEAR BRANCH

3 MECH

100 virat 05-NOV-84

rcb@gmail.com

4 Civil

SQL> select *from technicalskills;

PROGRAMMING_SKILLS WEB GRAPHICAL

COMPUTERSKILLS

sql html photoshop

msoffice

c java script free hand

powerpoint

java wordpress acrobat

spreadsheets

SQL> select *from learnedby;

SINCE SID

01-FEB-19 61

18-JUN-19 62

01-JAN-20 70

10-OCT-19 311

01-JAN-20 100

SQL> select *from certified_in;

PASSEDON SID

12-APR-19 61

19-AUG-19 62

01-MAR-20 70

30-NOV-19 311

13-FEB-20 100

DESCRIPTION:

The code describes about the technicalskills data retrieval of the student. The courses along with the technicalskills has been displayed of the student in this mini project.

THROUGH THE PROJECT:

This project helps to store data in a efficient way and it can be achieved through various sql commands and we can also store this for any future use and also we can save our data in a many different areas so we cannot lost all the data at once. The details cannot be lost so it is safer to use it .

IMPLEMENTATION

FRONT END PROGRAMS

1)insert student details:

package technicalskills;

import java.awt.Color;

import java.awt.FlowLayout;

import java.awt.GridLayout;

import java.awt.TextArea;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

//import java.beans.Statement;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

```
public class insert_students extends JFrame{

    /**
     *
     */
    private static final long serialVersionUID = 1L;
    JPanel jp1,jp2,jp3;
    Connection con;
    int i;
    java.sql.Statement stmt;
    JLabel sid;
    JLabel email,dob,year,branch,sname;
    JTextField s,e,d,y,b,sn;
    TextArea ta;
    JButton in;

    public insert_students()
    {
```

```
try  
{
```

```
Class.forName("oracle.jdbc.driver.OracleDriver");
```

```
con=DriverManager.getConnection("jdbc:oracle:thin:@l  
ocalhost:1521:xe","ajay","vasavi");
```

```
stmt=con.createStatement();
```

```
}
```

```
catch (Exception e) {
```

```
// TODO Auto-generated catch block
```

```
e.printStackTrace();
```

```
}
```

```
sid=new JLabel("studentid");
```

```
s=new JTextField(10);
```

```
sname=new JLabel("name of student");
```

```
sn=new JTextField(10);
```

```
dob=new JLabel("date of birth");
```

```
d=new JTextField(10);
email=new JLabel("student mailid");
e=new JTextField(10);
year=new JLabel("student year");
y=new JTextField(10);
branch=new JLabel("student branch");
b=new JTextField(10);
// email=new JLabel("student mailid");
//e=new JTextField(10);
ta=new TextArea(20,100);
in=new JButton("submit");
jp1=new JPanel(new GridLayout(5,1));
jp2=new JPanel(new FlowLayout());
jp3=new JPanel(new FlowLayout());
jp1.add(sid);
jp1.add(s);
jp1.add(sname);
jp1.add(sn);
jp1.add(dob);
jp1.add(d);
```

```
jp1.add(email);
jp1.add(e);
jp1.add(year);
jp1.add(y);
jp1.add(branch);
jp1.add(b);
// jp1.add(email);
// jp1.add(e);
jp2.add(in);
jp3.add(ta);
add(jp1);

// "insert into students values("'+ s.getText()
+ "','"+sn.getText()+"', '"+d.getText()+"', '"+e.getText()+"', '"+y.ge
tText()+"', '"+b.getText()+"')"
```

```
add(jp2);
add(jp3);
setVisible(true);
getContentPane().setBackground(Color.blue);
setSize(2000,1000);
setTitle("Enter following details:");
setLayout(new GridLayout(5,2));
```


pack();

in.addActionListener(new ActionListener() {

@Override

*public void actionPerformed(ActionEvent
argo) {*

*// TODO Auto-generated method
stub*

try {

*i=stmt.executeUpdate("insert
into students
values("+s.getText()+", "+sn.getText()+", "+d.getText()+", "+
e.getText()+", "+y.getText()+", "+b.getText()+")");*

} catch (SQLException e) {

*// TODO Auto-generated catch
block*

e.printStackTrace();

}

*ta.append("\n Inserted "+i+" rows
successfully");*

```
    }  
});
```

```
}
```

```
}
```

2)update student details:

```
package technicalskills;
```

```
import java.awt.Color;
```

```
import java.awt.FlowLayout;
```

```
import java.awt.GridLayout;
```

```
import java.awt.List;
```

```
import java.awt.TextArea;
```

```
import java.awt.event.ActionEvent;
```

```
import java.awt.event.ActionListener;
```

```
import java.awt.event.ItemEvent;
```

```
import java.awt.event.ItemListener;
```

```
//import java.beans.Statement;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.ResultSet;
```

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

```
import javax.swing.JButton;
```

```
import javax.swing.JFrame;
```

```
import javax.swing.JLabel;
```

```
import javax.swing.JPanel;
```

```
import javax.swing.JTextField;
```

```
public class update_students extends JFrame{
```

```
    /**
```

```
     *
```

```
    */
```

```
    private static final long serialVersionUID = 1L;
```

```
    JPanel jp1,jp2,jp3;
```

```
    Connection con;
```

```
    int i;
```

```
    java.sql.Statement stmt;
```

```
    JLabel sid;
```

JLabel email,dob,year,branch,sname;

TextField s,e,d,y,b,sn;

TextArea ta;

Button in;

List lis;

ResultSet rs;

String sel;

public update_students()

{

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:ORCL","ajay","vasavi");

stmt=con.createStatement();

}

```
catch (Exception e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}  
sid=new JLabel("studentid");  
s=new JTextField(10);  
sname=new JLabel("name of student");  
sn=new JTextField(10);  
dob=new JLabel("date of birth");  
d=new JTextField(10);  
email=new JLabel("student mailid");  
e=new JTextField(10);  
year=new JLabel("student year");  
y=new JTextField(10);  
branch=new JLabel("student branch");  
b=new JTextField(10);  
// email=new JLabel("student mailid");  
//e=new JTextField(10);  
ta=new TextArea(20,100);
```

```
in=new JButton("submit");
jp1=new JPanel(new FlowLayout());
jp2=new JPanel(new FlowLayout());
jp3=new JPanel(new FlowLayout());
jp1.add(sid);
jp1.add(s);
jp1.add(sname);
jp1.add(sn);
jp1.add(dob);
jp1.add(d);
jp1.add(email);
jp1.add(e);
jp1.add(year);
jp1.add(y);
jp1.add(branch);
jp1.add(b);
// jp1.add(email);
// jp1.add(e);
jp2.add(in);
jp3.add(ta);
```

```

        lis=new List();

        add(jp1);

        // "insert into students values("+ s.getText()
+""+sn.getText()+""+d.getText()+""+e.getText()+""+y.ge
tText()+""+b.getText()+"")"

        add(jp2);

        add(jp3);

        add(lis);

        try {

            rs=stmt.executeQuery("select sid from
students");

            while(rs.next()) {

                lis.add(rs.getString(1));

            }

        } catch (SQLException e) {

            // TODO Auto-generated catch block
            e.printStackTrace();

        }

        lis.addItemListener(new ItemListener() {

            @Override

```

```
public void itemStateChanged(ItemEvent  
    argo) {  
    // TODO Auto-generated method  
    stub  
    try {  
        sel=lis.getSelectedItemAt()  
        rs=stmt.executeQuery("select  
sname,email,dob,year,branch,sid from students where  
sid='"+lis.getSelectedItemAt()+"");  
        if(rs.next()) {  
            sn.setText(rs.getString(1));  
            e.setText(rs.getString(2));  
            d.setText(rs.getString(3));  
            y.setText(rs.getString(4));  
            b.setText(rs.getString(5));  
            s.setText(rs.getString(6));  
        }  
    } catch (SQLException e) {  
        // TODO Auto-generated catch  
        block  
        e.printStackTrace();  
    }  
}
```


}

});

in.addListener(new ActionListener() {

@Override

*public void actionPerformed(ActionEvent
argo) {*

*// TODO Auto-generated method
stub*

try {

*i=stmt.executeUpdate(" update
students set
sname='"+sn.getText()+"',email='"+e.getText()+"',year='"+y.ge
tText()+"',branch='"+b.getText()+"' where
sid='"+s.getText()+"");*

} catch (SQLException e) {

*// TODO Auto-generated catch
block*

e.printStackTrace();

}

```
ta.append("\n Updated "+i+"rows  
successfully");  
    }  
});
```

```
setVisible(true);  
getContentPane().setBackground(Color.blue);  
setSize(2000,1000);  
setTitle("Enter following details:");  
setLayout(new FlowLayout());  
pack();
```

```
}
```

```
}
```

3)delete student details:

```
package technicalskills;
```

```
import java.awt.Color;
```

```
import java.awt.FlowLayout;
```

```
import java.awt.GridLayout;
import java.awt.List;
import java.awt.TextArea;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
//import java.beans.Statement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
```

```
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;
```

```
public class delete_students extends JFrame{

    /**
     *
     */
    private static final long serialVersionUID = 1L;
    JPanel jp1,jp2,jp3;
    Connection con;
    int i;
    java.sql.Statement stmt;
    JLabel sid;
    JLabel email,dob,year,branch,sname;
    JTextField s,e,d,y,b,sn;
    TextArea ta;
    JButton in;
    List lis;
    ResultSet rs;
    String sel;

    public delete_students()
```

```
{  
    try  
    {
```

```
        Class.forName("oracle.jdbc.driver.OracleDriver");
```

```
        con=DriverManager.getConnection("jdbc:oracle:thin:@l  
ocalhost:1521:ORCL","ajay","vasavi");
```

```
        stmt=con.createStatement();
```

```
    }
```

```
    catch (Exception e) {
```

```
        // TODO Auto-generated catch block
```

```
        e.printStackTrace();
```

```
    }
```

```
    sid=new JLabel("studentid");
```

```
    s=new JTextField(10);
```

```
    sname=new JLabel("name of student");
```

```
    sn=new JTextField(10);
```

```
dob=new JLabel("date of birth");
d=new JTextField(10);
email=new JLabel("student mailid");
e=new JTextField(10);
year=new JLabel("student year");
y=new JTextField(10);
branch=new JLabel("student branch");
b=new JTextField(10);
// email=new JLabel("student mailid");
//e=new JTextField(10);
ta=new TextArea(20,100);
in=new JButton("submit");
jp1=new JPanel(new FlowLayout());
jp2=new JPanel(new FlowLayout());
jp3=new JPanel(new FlowLayout());
jp1.add(sid);
jp1.add(s);
jp1.add(sname);
jp1.add(sn);
jp1.add(dob);
```

```

        jp1.add(d);
        jp1.add(email);
        jp1.add(e);
        jp1.add(year);
        jp1.add(y);
        jp1.add(branch);
        jp1.add(b);
        //    jp1.add(email);
        //    jp1.add(e);
        jp2.add(in);
        jp3.add(ta);
        lis=new List();
        add(jp1);

        //    "insert into students values("'+ s.getText()
+ "','"+sn.getText()+"', '"+d.getText()+"', '"+e.getText()+"', '"+y.ge
tText()+"', '"+b.getText()+"')"
```

```

        add(jp2);
        add(jp3);
        add(lis);
        try {
```

```
students");
```

```
while(rs.next()) {
```

```
lis.add(rs.getString(1));
```

```
}
```

```
} catch (SQLException e) {
```

```
// TODO Auto-generated catch block
```

```
e.printStackTrace();
```

```
}
```

```
lis.addItemListener(new ItemListener() {
```

```
@Override
```

```
public void itemStateChanged(ItemEvent  
argo) {
```

```
// TODO Auto-generated method
```

```
stub
```

```
try {
```

```
sel=lis.getSelectedItem();
```

```
rs=stmt.executeQuery("select  
sname,email,dob,year,branch,sid from students where  
sid='"+lis.getSelectedItem()+"");
```

```
if(rs.next()) {
```



```

        sn.setText(rs.getString(1));
        e.setText(rs.getString(2));
        d.setText(rs.getString(3));
        y.setText(rs.getString(4));
        b.setText(rs.getString(5));
        s.setText(rs.getString(6));
    }
} catch (SQLException e) {
    // TODO Auto-generated catch
    e.printStackTrace();
}

});

in.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent
    argo) {

```

```

// TODO Auto-generated method
stub
try {

    i=stmt.executeUpdate("delete
from students where sid='"+s.getText()+"");
    } catch (SQLException e) {
        // TODO Auto-generated catch
        block
        e.printStackTrace();
    }
    ta.append("\n Deleted "+i+"rows
successfully");
}
});
```

```
setVisible(true);
getContentPane().setBackground(Color.blue);
setSize(2000,1000);
```

```
setTitle("Enter following details:");  
setLayout(new FlowLayout());  
pack();
```

```
}
```

```
}
```

4)Main Method:

```
package technicalskills;
```

```
import java.awt.Color;
```

```
import java.awt.FlowLayout;
```

```
import java.awt.event.ActionEvent;
```

```
import java.awt.event.ActionListener;
```

```
import javax.swing.JFrame;
```

```
import javax.swing.JMenu;
```

```
import javax.swing.JMenuBar;
```

```
import javax.swing.JMenuItem;
```

```

public class firstframe extends JFrame {

    /**
     *
     */
    // private static final long serialVersionUID = 1L;
    JMenuBar mnubar;
    JMenu m1,m2,m3,m4,m5;
    JMenuItem
in1,in2,in3,in4,in5,up1,up2,up3,up4,up5,dl1,dl2,dl3,dl4,dl5;
    public firstframe() {
        mnubar=new JMenuBar();
        m1=new JMenu("students");
        m2=new JMenu("technicalskills");
        m3=new JMenu("courses");
        m4=new JMenu("learnedby");
        m5=new JMenu("certifiedin");
        in1=new JMenuItem("Insert");
        up1=new JMenuItem("Update");
        dl1=new JMenuItem("Delete");
    }
}

```

```
        getContentPane().setBackground(Color.blue);
        setVisible(true);
        setSize(500,400);
        setTitle("Technicalskills data retrival");

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new FlowLayout());
        setJMenuBar(mnubar);
        mnubar.add(m1);
        mnubar.add(m2);
        mnubar.add(m3);
        mnubar.add(m4);
        mnubar.add(m5);
        m1.add(in1);
        m1.add(up1);
        m1.add(dl1);
        in1.addActionListener(new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent e) {
```

```

        // TODO Auto-generated method stub
        new insert_students();
        dispose();
    }
});

up1.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent
argo) {

        // TODO Auto-generated method stub
        new update_students();

    }
});

dl1.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub

```

```

        new delete_students();

    }

});

}

public static void main(String a[]) {
    new firstframe();
}

}

```

Thus, the connection from Java to Oracle database is performed and therefore, can be used for updating tables in the database directly.

SOFTWARE USED:

*Java Eclipse, Oracle 11g Database, Java SE version 7, SQL*Plus.*

Java AWT:

Java AWT (Abstract Window Toolkit) is an API to develop GUI or window-based applications in java.

Java AWT components are platform-dependent i.e. components are displayed according to the view of operating system.

AWT is heavyweight i.e. its components are using the resources of OS. The java.awt package provides classes for AWT API such as TextField, Label, TextArea, RadioButton, CheckBox, Choice, List etc.

SQL:

Structure Query Language(SQL) is a database query language used for storing and managing data in Relational DBMS. SQL was the first commercial language introduced for E.F Codd's Relational model of database. Today almost all RDBMS (MySQL, Oracle, Infomix, Sybase, MS Access) use SQL as the standard database query language. SQL is used to perform all types of data operations in RDBMS.






GITHUB LINK:

<https://github.com/bandaruajaybabu/Technical-skills-data-retrieval-DBMS>

FOLDER STRUCTURE:

This project contains a folder named src in which it has 5 different folders for different purposes each folder has 3 codes such as to make insert, delete, update. By this we can

navigate easily to reach code and we can make many changes as we can want easily.

Name	Date modified	Type	Size
 .settings	4/29/2020 7:19 PM	File folder	
 bin	4/30/2020 9:17 PM	File folder	
 src	4/29/2020 7:22 PM	File folder	
 .classpath	4/30/2020 4:17 PM	CLASSPATH File	1 KB
 .project	4/29/2020 7:19 PM	PROJECT File	1 KB

TESTING

The program executes three basic operations those are insert update and delete on 5 different tables. Along with this, it also has an output column which gives information about how many rows have been edited. Errors syntactical or exceptional will be shown if occurred.

HOME PAGE:

1)The home page represents the various options about the data retrieval of technical skills.



2)The second step checks the attributes present in the students table and allows to select an option and glance the details of it.



INSERT STUDENT DETAILS:

3) *The insert table allows us to insert a new student ID into the table along with the student name, DOB, email, year and branch. If there are no errors the student ID will be inserted successfully.*

```
SQL Plus
SQL> select * from students;

SID          SNAME          DOB
-----
EMAIL
-----
YEAR BRANCH
-----
66          badrinath       09-MAY-01
badri@gmail.com
2 it
62          ajay           20-DEC-00
ajay@gmail.com
2 it
SID          SNAME          DOB
-----
EMAIL
-----
YEAR BRANCH
-----
61          abhiraj        14-DEC-01
abhiraj123@gmail.com
2 it
SQL>
```

Enter following details:

studentid 70	name of student harsha	date of birth 1-feb-2001	student mailid harsha@gmail.com	student year 2	student branch 0	submit
--------------	------------------------	--------------------------	---------------------------------	----------------	------------------	--------

inserted rows successfully

SQL Plus

```
SQL> select * from students;
```

SID	SNAME	DOB
70	harsha	01-FEB-01
66	badrinath	09-MAY-01
62	ajay	20-DEC-00
61	abhiraj	14-DEC-01

```
SQL>
```

UPDATE STUDENT DETAILS:

4) This step allows to update the table of the existing details; we should select the student ID and then need to update the details.

Enter following details:

studentid: 70	name of student: harsha lohith	date of birth: 001-02-01 00:00:00	student mailid: harsha@gmail.com	student year: 2	student branch: 5	submit
---------------	--------------------------------	-----------------------------------	----------------------------------	-----------------	-------------------	--------

Updated Rows successfully

61
62
63
70

```
SQL Plus
SID          SNAME          DOB
-----
EMAIL
YEAR BRANCH
-----
61
abhiraj123@gmail.com    abhiraj    14-DEC-01
2 it

SQL> select * from students;
SID          SNAME          DOB
-----
EMAIL
YEAR BRANCH
-----
70
harsha@gmail.com       harsha lohith    01-FEB-01
2 it

66
badrinath@gmail.com    badrinath    09-MAY-01
2 it

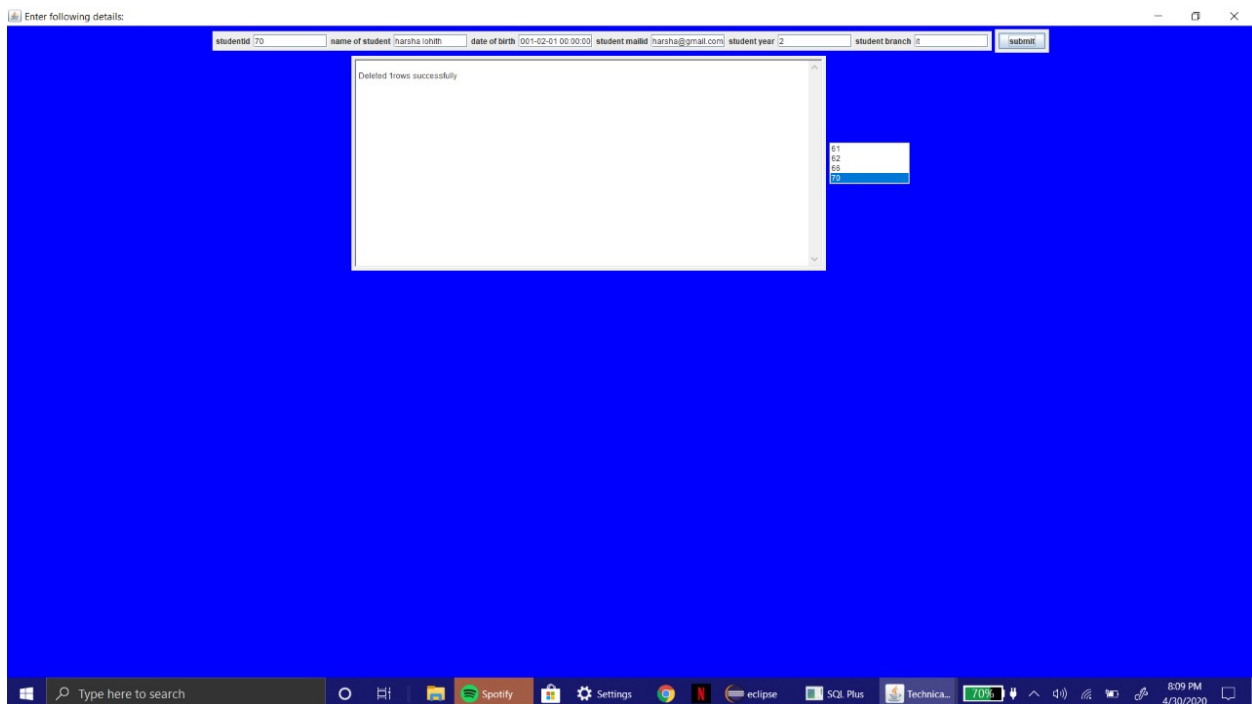
SID          SNAME          DOB
-----
EMAIL
YEAR BRANCH
-----
62
ajay@gmail.com         ajay          20-DEC-00

61
abhiraj123@gmail.com    abhiraj    14-DEC-01

SID          SNAME          DOB
-----
EMAIL
YEAR BRANCH
-----
2 it
```

DELETE STUDENT DETAILS:

5) This step allows us to delete the row which is existing and then select the row and press delete then it automatically deletes the selected student ID from the table.



```
SQL Plus

SQL> select * from students;

SID          SNAME          DOB
-----
EMAIL
-----
YEAR BRANCH
-----
66          badrinath          09-MAY-01
badri@gmail.com
2 it

62          ajay              20-DEC-00
ajay@gmail.com
2 it

SID          SNAME          DOB
-----
EMAIL
-----
YEAR BRANCH
-----
61          abhiraj          14-DEC-01
abhiraj123@gmail.com
2 it

SQL>
```

EXCEPTIONS:

6) This step gives a sql message when there is an error when the size of the attribute is exceeded.



DISCUSSIONS and FUTURE WORK:

Data retrieval isn't just about knowing where data resides. It also requires knowledge of health data attributes, including data definitions, value sets, and other administrative and clinical coded content.

REFERENCES:

<https://docs.oracle.com/javase/7/docs/api/>

<https://www.javatpoint.com/dbms-tutorial>

<https://www.techopedia.com/definition/30140/data-retrieval>

<http://www.ebooks-for-all.com/bookmarks/detail/Database-Management-Systems/onecat/0.html>