JAVA SWING BASED - AIR LINES QUALITY AND INFORMATION MANAGEMENT SYSTEM-SQL CONNECTIVITY USING JDBC

A

Report

Submitted in partial fulfilment of the
Requirements for the award of the Degree of
BACHELOR OF TECHNOLOGY
IN

INFORMATION TECHNOLOGY

By

AKSHITHA NAMPALLY <1602-20-737-004>

Under the Guidance of

B. Leelavathy

Department of Information Technology

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31

2021-2022

BONAFIDE CERTIFICATE

This to Certify that the project report titled **"SYNTAX AND DESCRIPTION IN 3PL'S"** project work of Ms. Akshitha Nampally bearing Roll.no:1602-20-737-004 who carried out this project under my supervision in the IV semester for the academic year 2021-2022.

Signature *Signature*

external examiner

internal examiner

ROLL NO: 1602-20-737-004
NAME: Akshitha Nampally. [2]

DATATYPES AND THEIR SYNTAXES IN 3PL'S

ASSIGNMENT-1

N. Akshitha

1602-20-737-004

ABSTRACT:

Data types and their syntaxes in 3 programming languages aims for the description of the datatypes of different programming languages when selected any datatype it should display the description and syntax of the database. That means here when we select any programming language it should display the respective language and the data types and their syntaxes and the required description

REQUIREMENT ANALYSIS

List of Tables**:**

- Language
- High-level language
- Medium-level language
- Low-level language

List of Attributes with their Domain Types:

Language:

- L_NAME VARCHAR2(20)
- P_ID NUMBER(20)
- J_ID NUMBER(20)
- C_ID NUMBER(20)

High-level language: (for python)

- DATA_TYPE VARCHAR2(20)
- SYNTAX VARCHAR2(20)

- DESCRIPTION VARCHAR2(20)
- P_ID NUMBER(10)

Medium-level language: (for C)

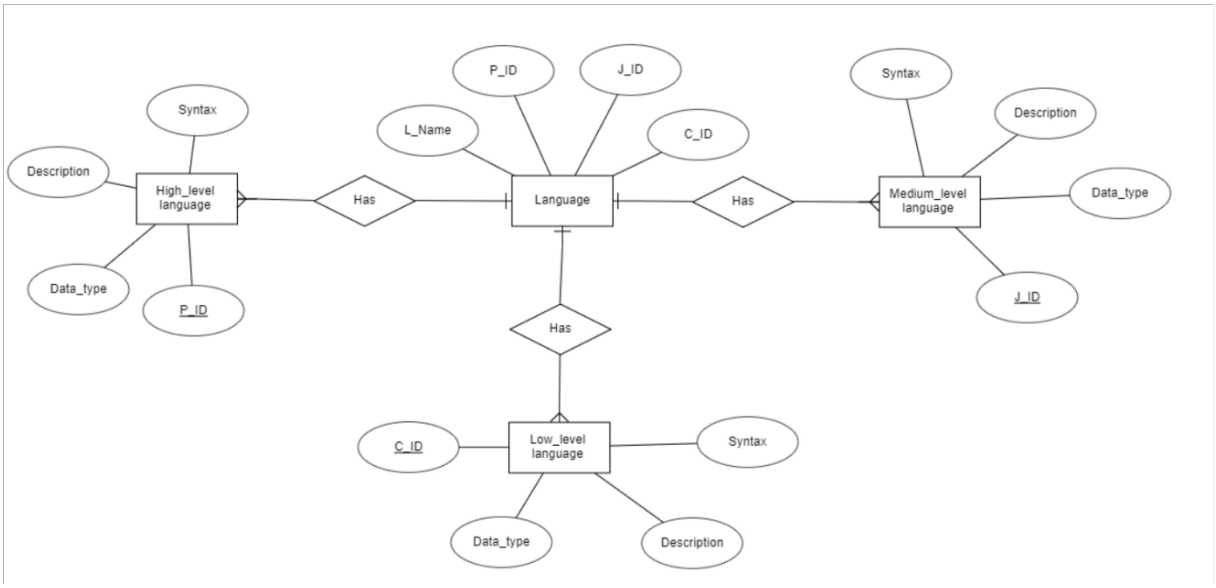
- DATA_TYPE VARCHAR2(20)
- SYNTAX VARCHAR2(20)
- DESCRIPTION VARCHAR2(20)
- J_ID NUMBER(10)

Low-level language: (for assembly language)

- DATA_TYPE VARCHAR2(20)
- SYNTAX VARCHAR2(20)
- DESCRIPTION VARCHAR2(20)
- C_ID NUMBER(10)

DESIGN

ENTITY RELATIONSHIP DIAGRAM



DDL COMMANDS:

1. Creating table for language:

Create table Language(

P_ID VARCHAR2(20),

J_ID VARCHAR2(20),

C_ID VARCHAR2(20));

OUTPUT:

```
SQL> create table Language(
  2  P_ID VARCHAR2(20),
  3  J_ID VARCHAR2(20),
  4  C_ID VARCHAR2(20));

Table created.

SQL> desc Language;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 P_ID                                               VARCHAR2(20)
 J_ID                                               VARCHAR2(20)
 C_ID                                               VARCHAR2(20)

SQL>
```

2. Creating table for High_level_language:

Create table high_level_language(

Data_type varchar2(20),

Syntax varchar2(20),

Description varchar2(20),

P_ID number(10));

Output:

```
SQL> create table HIGH_LEVEL_LANGUAGE(
  2  DATA_TYPE VARCHAR2(20),
  3  SYNTAX VARCHAR2(20),
  4  DESCRIPTION VARCHAR2(20),
  5  P_ID NUMBER(10));

Table created.

SQL> DESC HIGH_LEVEL_LANGUAGE;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DATA_TYPE                                          VARCHAR2(20)
 SYNTAX                                             VARCHAR2(20)
 DESCRIPTION                                        VARCHAR2(20)
 P_ID                                               NUMBER(10)
```

Making P_ID as primary key

Alter table high_level_language add primary key(P_ID);

Output:

```
SQL> ALTER TABLE HIGH_LEVEL_LANGUAGE ADD PRIMARY KEY(P_ID);

Table altered.

SQL> DESC HIGH_LEVEL_LANGUAGE;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DATA_TYPE                                          VARCHAR2(20)
 SYNTAX                                             VARCHAR2(20)
 DESCRIPTION                                        VARCHAR2(20)
 P_ID                                      NOT NULL NUMBER(10)
```

3. Creating table for Medium_level_language:

Create table medium_level_language(

Data_type varchar2(20),

Syntax varchar2(20),

Description varchar2(20),

J_ID number(10));

Output :

```
SQL> CREATE TABLE MEDIUM_LEVEL_LANGUAGE(
  2  DATA_TYPE VARCHAR2(20),
  3  SYNTAX VARCHAR2(20),
  4  DESCRIPTION VARCHAR2(20),
  5  J_ID NUMBER(10));

Table created.

SQL> DESC MEDIUM_LEVEL_LANGUAGE;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DATA_TYPE                                          VARCHAR2(20)
 SYNTAX                                             VARCHAR2(20)
 DESCRIPTION                                        VARCHAR2(20)
 J_ID                                               NUMBER(10)
```

Making J_ID as primary key

Alter table medium_level_language add primary key(J_ID);

Output:

```
SQL> DESC MEDIUM_LEVEL_LANGUAGE;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DATA_TYPE                                          VARCHAR2(20)
 SYNTAX                                             VARCHAR2(20)
 DESCRIPTION                                        VARCHAR2(20)
 J_ID                                               NUMBER(10)

SQL> ALTER TABLE MEDIUM_LEVEL_LANGUAGE ADD PRIMARY KEY(J_ID);

Table altered.

SQL> DESC MEDIUM_LEVEL_LANGUAGE;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DATA_TYPE                                          VARCHAR2(20)
 SYNTAX                                             VARCHAR2(20)
 DESCRIPTION                                        VARCHAR2(20)
 J_ID                                      NOT NULL NUMBER(10)
```

4. Creating table for Low_level_language:

Create table low_level_language(
Data_type varchar2(20),

Syntax varchar2(20),

Description varchar2(20),

C_ID number(10));

Outputs:

```
SQL> CREATE TABLE LOW_LEVEL_LANGUAGE(
  2  DATA_TYPE VARCHAR2(20),
  3  SYNTAX VARCHAR2(20),
  4  DESCRIPTION VARCHAR2(20),
  5  C_ID NUMBER(10));

Table created.

SQL> DESC LOW_LEVEL_LANGUAGE;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DATA_TYPE                                          VARCHAR2(20)
 SYNTAX                                             VARCHAR2(20)
 DESCRIPTION                                        VARCHAR2(20)
 C_ID                                               NUMBER(10)
```

Making C_ID as primary key

Alter table low_level_language add primary key(C_ID);

Outputs:

```
SQL> ALTER TABLE LOW_LEVEL_LANGUAGE ADD PRIMARY KEY(C_ID);

Table altered.

SQL> DESC LOW_LEVEL_LANGUAGE;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DATA_TYPE                                          VARCHAR2(20)
 SYNTAX                                             VARCHAR2(20)
 DESCRIPTION                                        VARCHAR2(20)
 C_ID                                      NOT NULL NUMBER(10)
```

DML COMMANDS:

1.

Inserting values into Language :

Insert into language (1,2,3);

Outputs:

```
SQL> insert into language values(1,2,3);
```

```
SQL> SELECT * FROM LANGUAGE;

P_ID                 J_ID                 C_ID
-------------------- -------------------- --------------------
1                    2                    3

SQL>
```

2.

Inserting values into table high_level_language:

insert into high_level_language values('&data_type','&syntax','&description',1);

Outputs:

```
SQL> select * from high_level_language;

DATA_TYPE           SYNTAX              DESCRIPTION               P_ID
------------------- ------------------- ------------------- ----------
integer             int()               returns integer              1

SQL> insert into high_level_language values('&data_type','&syntax','&description',1);
Enter value for data_type: float
Enter value for syntax: float()
Enter value for description: returns decimal
old   1: insert into high_level_language values('&data_type','&syntax','&description',1)
new   1: insert into high_level_language values('float','float()','returns decimal',1)

1 row created.

SQL> /
Enter value for data_type: character
Enter value for syntax: char()
Enter value for description: returns character
old   1: insert into high_level_language values('&data_type','&syntax','&description',1)
new   1: insert into high_level_language values('character','char()','returns character',1)

1 row created.

SQL> commit;

Commit complete.

SQL> select * from high_level_language;

DATA_TYPE           SYNTAX              DESCRIPTION               P_ID
------------------- ------------------- ------------------- ----------
integer             int()               returns integer              1
float               float()             returns decimal              1
character           char()              returns character            1

SQL>
```

3.

Inserting values into medium_level_language:

SQL> insert into medium_level_language values('&data_type','&syntax','&description',2);

Outputs:

```
SQL> desc medium_level_language;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DATA_TYPE                                          VARCHAR2(20)
 SYNTAX                                             VARCHAR2(20)
 DESCRIPTION                                        VARCHAR2(20)
 J_ID                                               NUMBER(10)

SQL> insert into medium_level_language values('&data_type','&syntax','&description',2);
Enter value for data_type: integer
Enter value for syntax: int
Enter value for description: returns integer
old   1: insert into medium_level_language values('&data_type','&syntax','&description',2)
new   1: insert into medium_level_language values('integer','int','returns integer',2)

1 row created.

SQL> /
Enter value for data_type: float
Enter value for syntax: float
Enter value for description: returns decimal
old   1: insert into medium_level_language values('&data_type','&syntax','&description',2)
new   1: insert into medium_level_language values('float','float ','returns decimal',2)

1 row created.

SQL> /
Enter value for data_type: character
Enter value for syntax: char
Enter value for description: returns character
old   1: insert into medium_level_language values('&data_type','&syntax','&description',2)
new   1: insert into medium_level_language values('character','char','returns character',2)

1 row created.

SQL> commit;

Commit complete.

SQL> select * from medium_level_language;

DATA_TYPE           SYNTAX              DESCRIPTION               J_ID
------------------- ------------------- ------------------- ----------
integer             int                 returns integer              2
float               float               returns decimal              2
character           char                returns character            2
```

4.

Inserting the values into table low_level_language:

SQL> insert into low_level_language values('&data_type','&syntax','&description',3);

Outputs:

```
SQL> desc low_level_language;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DATA_TYPE                                          VARCHAR2(20)
 SYNTAX                                             VARCHAR2(20)
 DESCRIPTION                                        VARCHAR2(20)
 C_ID                                               NUMBER(10)

SQL> insert into low_level_language values('&data_type','&syntax','&description',3);
Enter value for data_type: byte
Enter value for syntax: BYTE
Enter value for description: returns bytes
old   1: insert into low_level_language values('&data_type','&syntax','&description',3)
new   1: insert into low_level_language values('byte','BYTE','returns bytes',3)

1 row created.

SQL> /
Enter value for data_type: sbyte
Enter value for syntax: SBYTE
Enter value for description: returns bytes
old   1: insert into low_level_language values('&data_type','&syntax','&description',3)
new   1: insert into low_level_language values('sbyte','SBYTE','returns bytes',3)

1 row created.

SQL> /
Enter value for data_type: word
Enter value for syntax: WORD
Enter value for description: returns words
old   1: insert into low_level_language values('&data_type','&syntax','&description',3)
new   1: insert into low_level_language values('word','WORD','returns words',3)

1 row created.

SQL> select * from low_level_language;

DATA_TYPE            SYNTAX               DESCRIPTION               C_ID
-------------------- -------------------- -------------------- ----------
byte                 BYTE                 returns bytes                3
sbyte                SBYTE                returns bytes                3
word                 WORD                 returns words                3
```

DATATYPES AND THEIR SYNTAXES IN 3PL'S

ASSIGNMENT -2

N. Akshitha

1602-20-737-004

Implementation:

The front end has been developed using Java Swings. Swing is a Java Foundation Classes JFC library and an extension of the Abstract Window Toolkit. Swing offers much-improved functionality over AWT. Swing is entirely written in Java; hence it is platform independent and lightweight. It also supports pluggable look and feel.

So, to improve the performance and make the UI look more attractive, I have chosen to use Swings.

HIGH_LEVEL_LANGUAGE TABLE :

Code:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;


class HomePageUI extends JFrame implements ActionListener
{
        High_level_languageUI ob1;
        Medium_level_languageUI     ob2;
        Low_level_languageUI ob3;


        JButton submit,modify,delete,m1,m2,m3;
        JPanel p1,p2,p3,pb;
        JMenuBar mb;
```

```
public HomePageUI()
{
        setSize(600,550);
        setLayout(null);
        setVisible(true);
        setTitle("Programming Languages");

        ob1 = new High_level_languageUI();
        ob2 = new Medium_level_languageUI();
        ob3 = new Low_level_languageUI();

        createPanels();
        createMenu();
        createButtons();
        addComponents();
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

void createPanels()
{
        p1 = ob1.p;
        p2 = ob2.p;
        p3 = ob3.p;

        pb = new JPanel(new FlowLayout(FlowLayout.CENTER,50,0));
        pb.setBounds(0,400,600,150);
}
```

```
void createMenu()
{
        mb = new JMenuBar();

        m1 = new JButton("High_level_languageUI");
        m1.setFocusable(false);

        m2 = new JButton("Medium_level_languageUI");
        m2.setFocusable(false);


        m3 = new JButton("Low_level_languageUI");
        m3.setFocusable(false);


        m1.addActionListener(this);
        m2.addActionListener(this);
        m3.addActionListener(this);

        mb.add(m1);
        mb.add(m2);
        mb.add(m3);

}

public void actionPerformed(ActionEvent e)
{
        remove(p1);
        remove(p2);
```

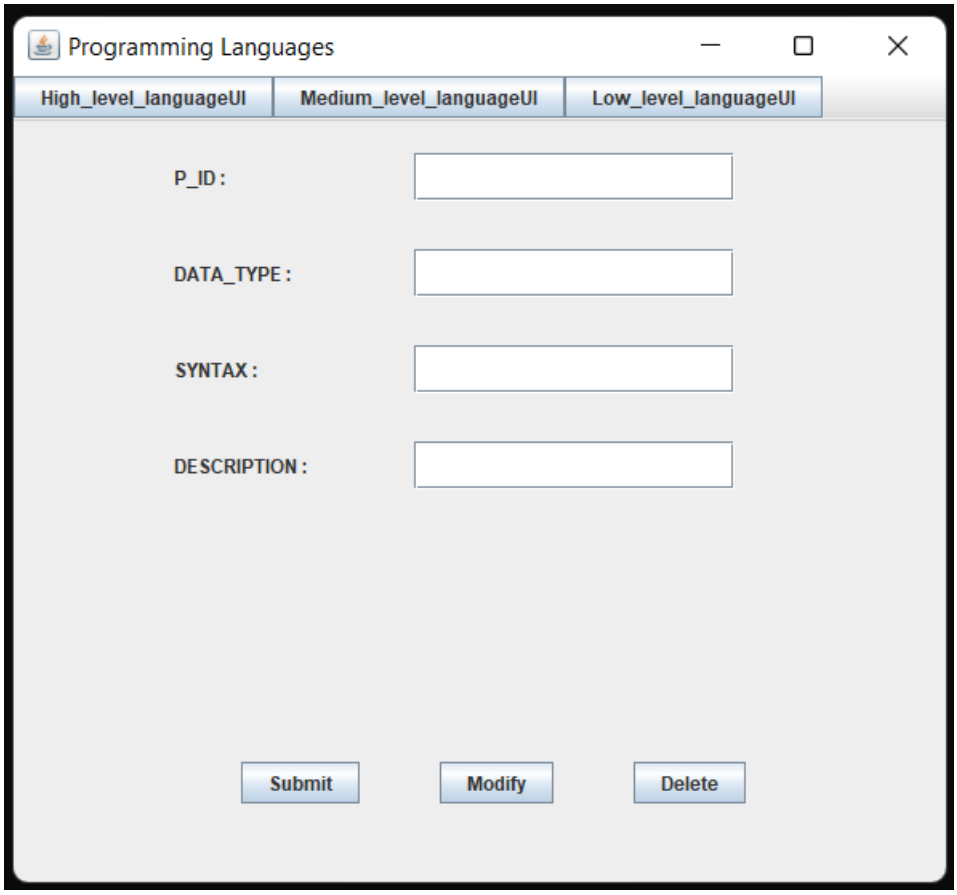
```
                remove(p3);

                if(e.getSource()==m1)
                        add(p1);

                else if(e.getSource()==m2)
                        add(p2);

                else
                        add(p3);
        }

        void createButtons()
        {
                submit = new JButton("Submit");
                submit.addActionListener(new ActionListener()
                {
                        public void actionPerformed(ActionEvent e)
                        {
                                JOptionPane.showMessageDialog(new JFrame(),"Successfully
Inserted!","NOTICE",JOptionPane.INFORMATION_MESSAGE);
                        }
                });


                modify = new JButton("Modify");
                modify.addActionListener(new ActionListener()
                {
                        public void actionPerformed(ActionEvent e)
                        {
```

```
                    JOptionPane.showMessageDialog(new JFrame(),"Successfully Modified!","NOTICE",JOptionPane.INFORMATION_MESSAGE);
                }
            });


            delete = new JButton("Delete");
            delete.addActionListener(new ActionListener()
            {
                public void actionPerformed(ActionEvent e)
                {
                    JOptionPane.showMessageDialog(new JFrame(),"Successfully Deleted!","NOTICE",JOptionPane.INFORMATION_MESSAGE);
                }
            });

            pb.add(submit);
            pb.add(modify);
            pb.add(delete);
        }

        void addComponents()
        {
            add(p1);
            add(pb);
            setJMenuBar(mb);
        }

        public static void main(String a[])
        {
```

```
                    new HomePageUI();
            }
}


import javax.swing.*;
class High_level_languageUI
{
JTextField t1,t2,t3,t4;
JLabel l1,l2,l3,l4;
JPanel p;
public High_level_languageUI()
{
createComponents();
addComponents();
}
void createComponents()
{
t1 = new JTextField();
t1.setBounds(250,20,200,30);
t2 = new JTextField();
t2.setBounds(250,80,200,30);
t3 = new JTextField();
t3.setBounds(250,140,200,30);
t4 = new JTextField();
t4.setBounds(250,200,200,30);
l1 = new JLabel("P_ID : ");
l1.setBounds(100,20,100,30);
l2 = new JLabel("DATA_TYPE : ");
```

```
l2.setBounds(100,80,100,30);
l3 = new JLabel("SYNTAX : ");
l3.setBounds(100,140,100,30);
l4 = new JLabel("DESCRIPTION : ");
l4.setBounds(100,200,100,30);
p = new JPanel(null);
p.setBounds(0,0,600,400);
}
void addComponents()
{
p.add(l1);
p.add(t1);
p.add(l2);
p.add(t2);
p.add(l3);
p.add(t3);
p.add(l4);
p.add(t4);
}
}
```

Output:



MEDIUM_LEVEL_LANGUAGE TABLE :

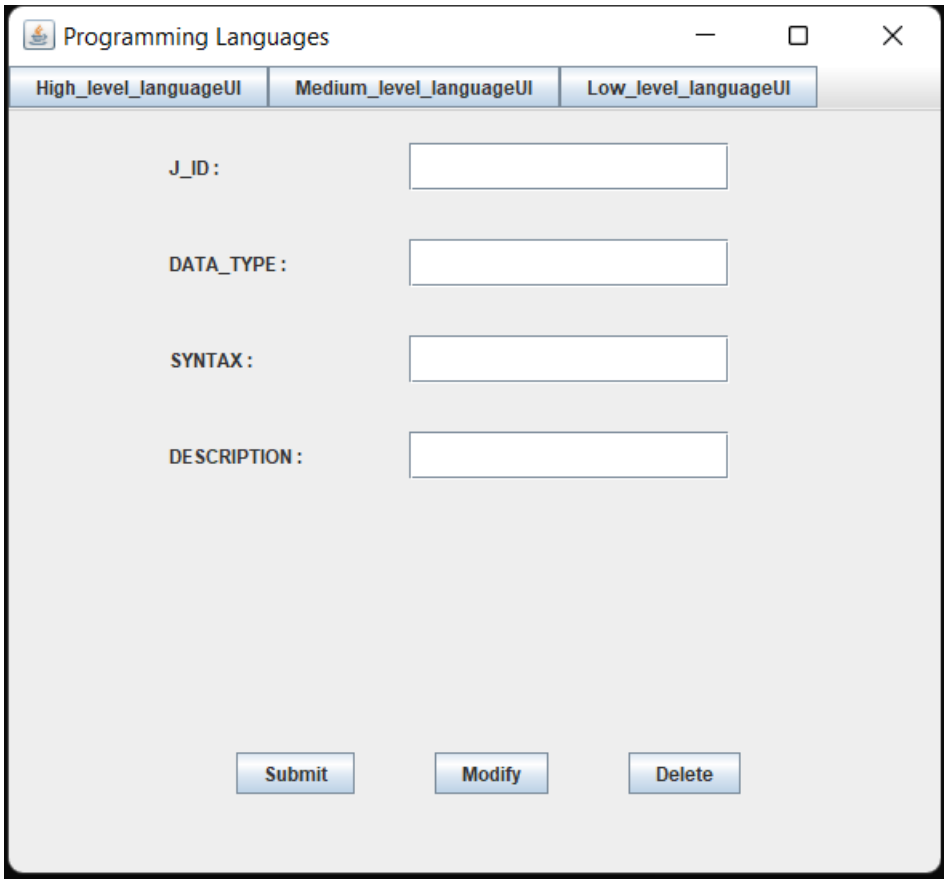
Code:

```
import javax.swing.*;
class Medium_level_languageUI
{
JTextField t1,t2,t3,t4;
JLabel l1,l2,l3,l4;
JPanel p;
public Medium_level_languageUI()
{
createComponents();
addComponents();
```

```
}
void createComponents()
{
t1 = new JTextField();
t1.setBounds(250,20,200,30);
t2 = new JTextField();
t2.setBounds(250,80,200,30);
t3 = new JTextField();
t3.setBounds(250,140,200,30);
t4 = new JTextField();
t4.setBounds(250,200,200,30);
l1 = new JLabel("J_ID : ");
l1.setBounds(100,20,100,30);
l2 = new JLabel("DATA_TYPE : ");
l2.setBounds(100,80,100,30);
l3 = new JLabel("SYNTAX : ");
l3.setBounds(100,140,100,30);
l4 = new JLabel("DESCRIPTION : ");
l4.setBounds(100,200,100,30);
p = new JPanel(null);
p.setBounds(0,0,600,400);
}
void addComponents()
{
p.add(l1);
p.add(t1);
p.add(l2);
p.add(t2);
p.add(l3);
```

```
p.add(t3);
p.add(l4);
p.add(t4);
}
}
```

Output:



LOW_LEVEL_LANGUAGE TABLE :

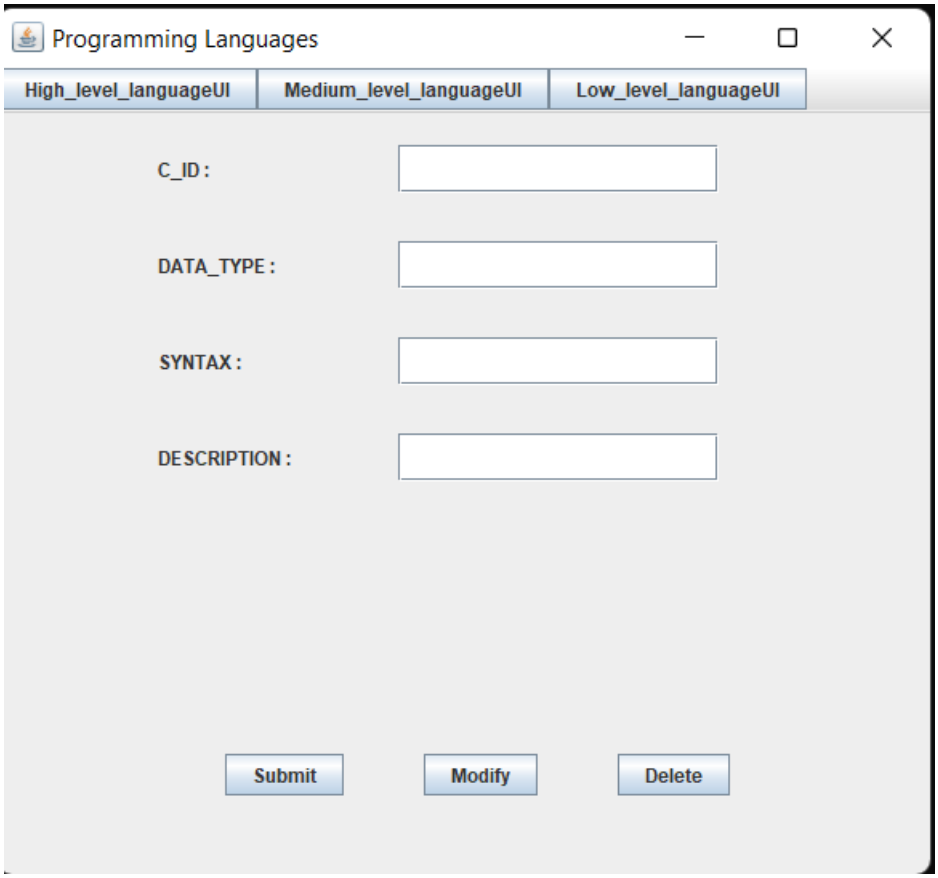
Code:

```
import javax.swing.*;
class Low_level_languageUI
```

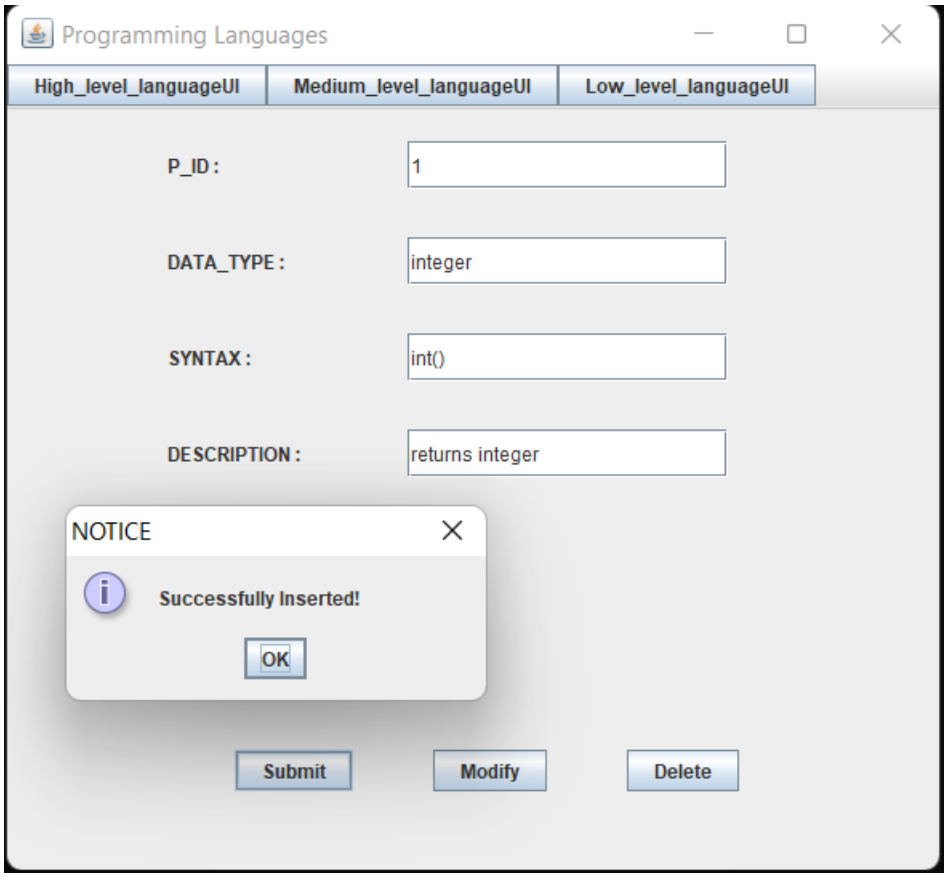
```
{
JTextField t1,t2,t3,t4;
JLabel l1,l2,l3,l4;
JPanel p;
public Low_level_languageUI()
{
createComponents();
addComponents();
}
void createComponents()
{
t1 = new JTextField();
t1.setBounds(250,20,200,30);
t2 = new JTextField();
t2.setBounds(250,80,200,30);
t3 = new JTextField();
t3.setBounds(250,140,200,30);
t4 = new JTextField();
t4.setBounds(250,200,200,30);
l1 = new JLabel("C_ID : ");
l1.setBounds(100,20,100,30);
l2 = new JLabel("DATA_TYPE : ");
l2.setBounds(100,80,100,30);
l3 = new JLabel("SYNTAX : ");
l3.setBounds(100,140,100,30);
l4 = new JLabel("DESCRIPTION : ");
l4.setBounds(100,200,100,30);
p = new JPanel(null);
p.setBounds(0,0,600,400);
```

```
}
void addComponents()
{
p.add(l1);
p.add(t1);
p.add(l2);
p.add(t2);
p.add(l3);
p.add(t3);
p.add(l4);
p.add(t4);
}
}
```

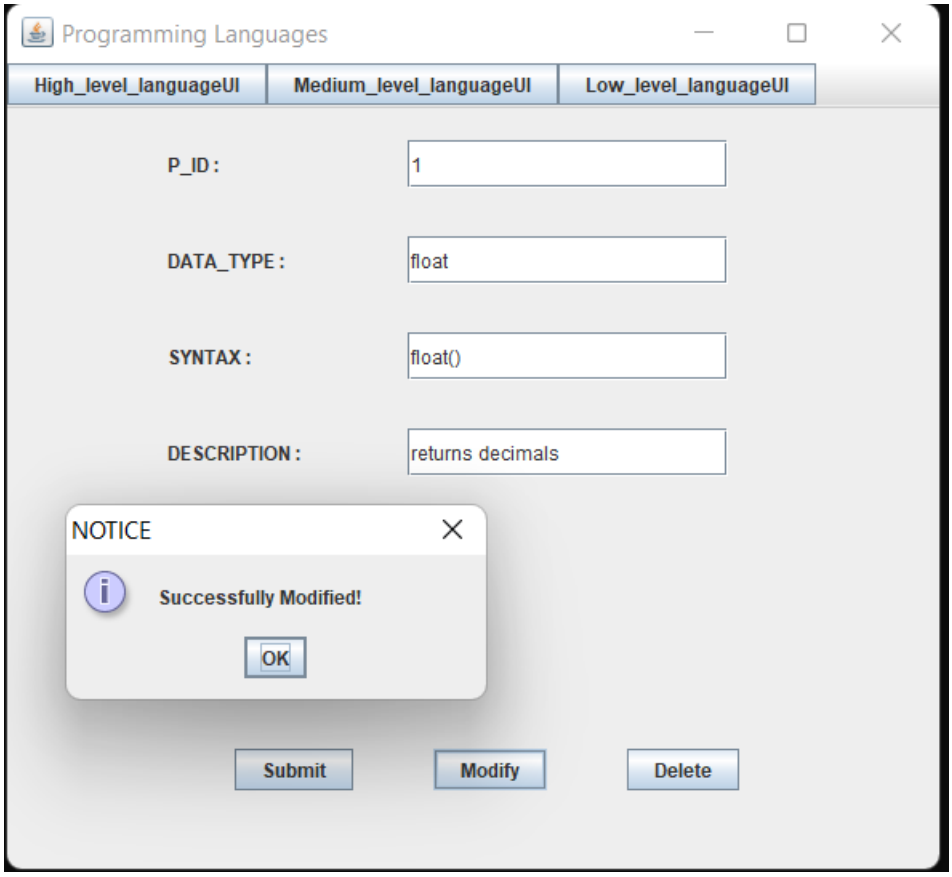
Output:


Programming Languages
High_level_languageUI
Medium_level_languageUI
Low_level_languageUI
C_ID :
DATA_TYPE :
SYNTAX :
DESCRIPTION :
Submit
Modify
Delete

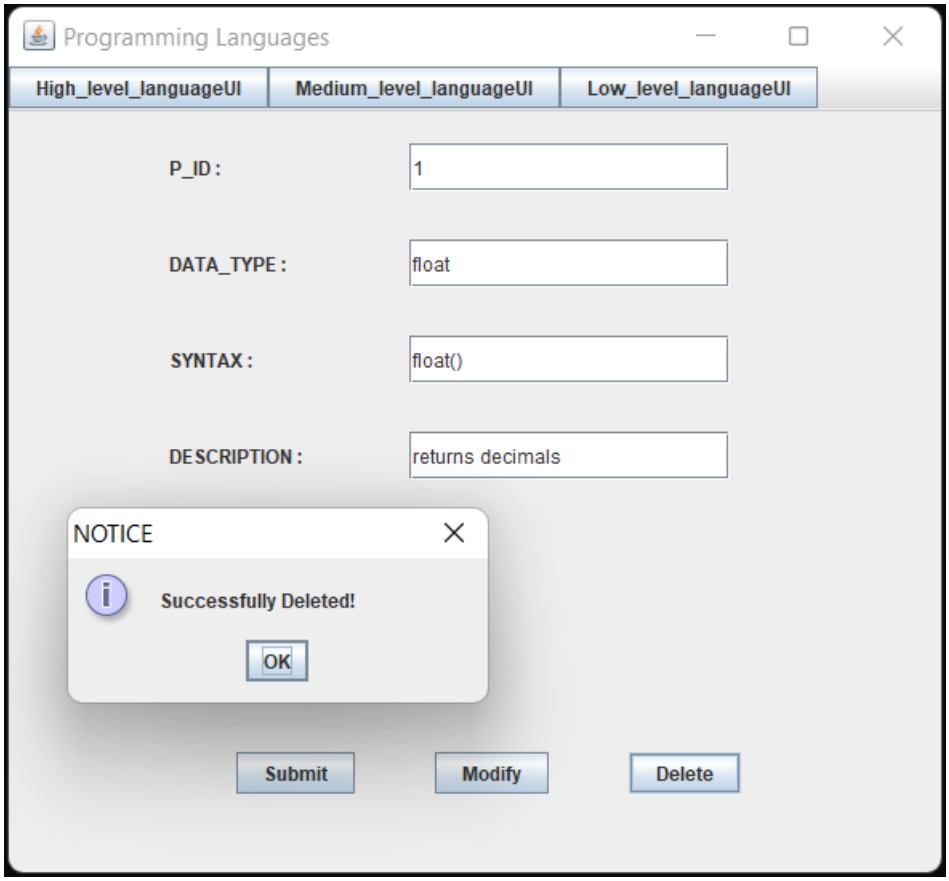
INSERTING:



MODIFYING:


Programming Languages
High_level_languageUI
Medium_level_languageUI
Low_level_languageUI
P_ID :
1
DATA_TYPE :
float
SYNTAX :
float()
DESCRIPTION :
returns decimals
NOTICE
Successfully Modified!
OK
Submit
Modify
Delete

DELETING:


Programming Languages
High_level_languageUI
Medium_level_languageUI
Low_level_languageUI
P_ID :
1
DATA_TYPE :
float
SYNTAX :
float()
DESCRIPTION :
returns decimals
NOTICE
Successfully Deleted!
OK
Submit
Modify
Delete

IMPLEMENTATION :

Front end programs and its connectivity Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database and is oriented towards relational databases. The connection to the database can be performed using Java programming (JDBC API) as:

```
import java.sql.*;

public class TrialConnect{

	public static void main(String[] args){

		try{

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

			Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","akshitha","vasavi");

			Statement stmt=con.createStatement();

			ResultSet rs=stmt.executeQuery("select * from sailors");

			while(rs.next())

				System.out.println(rs.getInt(1)+"  "+rs.getString(2));

			con.close();

		}

		catch(Exception e){

			System.out.println(e);

		}

	}

}
```

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

Program:

MainPage:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class HomePage extends JFrame{
        private JFrame frame= new JFrame();
        private JButton b1=new JButton("High_level_language");
        private JButton b2=new JButton("Medium_level_language");
        private JButton b3=new JButton("Low_level_language");
        //private JButton b4=new JButton("Severity Page");
        //private JLabel l1=new JLabel("DIRECT TO");
        private JMenuBar mBar;
   private JMenu mnuHelp;
   private JMenuItem abt;
        public HomePage(){
     frame.setTitle("Home Page");
     frame.setLayout(null);
     frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
     frame.setBounds(100,150,1000,400);
                Container c=frame.getContentPane();
                frame.getContentPane().add(b1);
                frame.getContentPane().add(b2);
                frame.getContentPane().add(b3);
                //frame.getContentPane().add(b4);
                //frame.getContentPane().add(l1);
                initializeMenuBar();
     frame.setJMenuBar(mBar);
                abt.addActionListener(new HelpMenuActionListener());
                JLabel label=new JLabel("DATATYPES AND THEIR SYNTAXES IN 3PL's");
                //JLabel label1=new JLabel();
     //label1.setIcon(new ImageIcon("C:/Users/sasid/Downloads/maize.jpg"));
     //Dimension size = label1.getPreferredSize();
```

```
label.setBounds(65,5,700,50);
label.setFont(new Font("Serif",Font.PLAIN,20));
		//label.setForeground(Color.BLUE);
		//l1.setBounds(670,5,700,50);
//l1.setFont(new Font("Serif",Font.PLAIN,20));
		//l1.setForeground(Color.BLUE);
//label1.setBounds(200,60, size.width,size.height);
		b1.setBounds(640,60,170,40);
		b1.setFont(new Font("Times New Roman",Font.BOLD,17));
		b2.setBounds(640,110,170,40);
		b2.setFont(new Font("Times New Roman",Font.BOLD,17));
		b3.setBounds(640,160,170,40);
		b3.setFont(new Font("Times New Roman",Font.BOLD,17));
		//b4.setBounds(640,210,170,40);
		//b4.setFont(new Font("Times New Roman",Font.BOLD,17));
b1.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent ae) {
				new High_level_language();
			}
		});
b2.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent ae) {
				new Medium_level_language();
			}
		});
b3.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent ae) {
				new Low_level_language();
			}
		});
		/* b4.addActionListener(new ActionListener() {
```

```
                public void actionPerformed(ActionEvent ae) {
                        new Severity();
                }
        });     */
                c.add(label);
                //c.add(label1);
                //frame.getContentPane().setBackground(Color.CYAN);
                frame.setVisible(true);
    }
        public void initializeMenuBar()
    {
                mBar=new JMenuBar();
                mnuHelp=new JMenu("Help");
                abt=new JMenuItem("About");
                mnuHelp.add(abt);
                mBar.add(mnuHelp);
    }
        private class HelpMenuActionListener implements ActionListener {
                public void actionPerformed(ActionEvent ae) {
                if(ae.getSource()==abt)
                {
                        String details;
                        details = "This project is about recording the datatypes of different
programming languages when selected any datatype it should display the description and syntax of
thr database"+"\n"+
            "It has 3 tables:"+"\n"+
            "1.High_level_language table with rows containing python Id as P_ID,data type as
DATA_TYPE ,its syntax as SYNTAX and its description as DESCRIPTION"+"\n"+
            "2.Medium_level_language table with rows containing java Id as J_ID,data type as
DATA_TYPE ,its syntax as SYNTAX and its description as DESCRIPTION"+"\n"+
            "3.Low_level_language table with rows containing cId as C_ID,data type as
DATA_TYPE, its syntax as SYNTAX and its description as DESCRIPTION ";
```

```
                                    JOptionPane.showMessageDialog(null,details,"INFORMATION",
JOptionPane.INFORMATION_MESSAGE);
                                }
                            }
                        }
        public static void main(String args[]){
                new HomePage();
}
}
```

TABLE : HIGH_LEVEL_LANGUAGE

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import java.util.*;
public class High_level_language implements ActionListener{
        private JFrame f=new JFrame("High_level_language");
        private JLabel l=new JLabel("");
        private JLabel l1=new JLabel("Enter datatype to insert");
        private JLabel l2=new JLabel("Enter datatype to update");
        //private JLabel l3=new JLabel("Enter datatype to delete");
        //private JLabel l4=new JLabel("Result");
        //private JLabel l5=new JLabel("DEFICIENCY");
        private JButton b1=new JButton("Insert");
        private JButton b2=new JButton("Update");
        //private JButton b3=new JButton("Delete");
        //private JButton b4=new JButton("Retrieve");
        private JTextField t1=new JTextField();
        private JTextField t2=new JTextField();
        //private JTextField t3=new JTextField();
        private JTextField t=new JTextField();
```

```
	private JTextArea t4=new JTextArea();
	private JScrollPane scrollBar=new
JScrollPane(t4,JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,JScrollPane.HORIZONTAL_SCROLLBAR_AL
WAYS);
	public High_level_language() {
		f.setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);
		f.setBounds(100,150,1000,400);
		Container c=f.getContentPane();
		f.getContentPane().add(l1);
		f.getContentPane().add(l2);
		//f.getContentPane().add(l3);
		//f.getContentPane().add(l4);
		//f.getContentPane().add(l5);
		f.getContentPane().add(b1);
		f.getContentPane().add(b2);
		//f.getContentPane().add(b3);
		//f.getContentPane().add(b4);
		f.getContentPane().add(t1);
		f.getContentPane().add(t2);
		//f.getContentPane().add(t3);
		f.getContentPane().add(t);
		f.getContentPane().add(scrollBar);
		scrollBar.setBounds(690,80,250,150);
		l.setBounds(20,30,50,50);
		l1.setBounds(60,80,250,30);
		l1.setOpaque(true);
		l1.setBackground(Color.WHITE);
		l2.setBounds(60,120,250,30);
		l2.setOpaque(true);
		l2.setBackground(Color.WHITE);
		//l3.setBounds(60,160,250,30);
		//l3.setOpaque(true);
```

```
                //l3.setBackground(Color.WHITE);
                //l4.setBounds(60,200,250,30);
                //l4.setOpaque(true);
                //l4.setBackground(Color.WHITE);
                //l5.setBounds(430,5,700,50);
//l5.setFont(new Font("Serif",Font.PLAIN,20));
                //l5.setForeground(Color.BLUE);
                b1.setBounds(570,80,100,30);
                b1.setFont(new Font("Times New Roman",Font.BOLD,17));
                b2.setBounds(570,120,100,30);
                b2.setFont(new Font("Times New Roman",Font.BOLD,17));
                //b3.setBounds(570,160,100,30);
                //b3.setFont(new Font("Times New Roman",Font.BOLD,17));
                //b4.setBounds(570,200,100,30);
                //b4.setFont(new Font("Times New Roman",Font.BOLD,17));
                t1.setBounds(330,80,220,30);
                t2.setBounds(330,120,220,30);
                //t3.setBounds(330,160,220,30);
                t.setBounds(330,200,220,30);
                t4.setEditable(false);
                b1.addActionListener(this);
                b2.addActionListener(this);
                //b3.addActionListener(this);
                //b4.addActionListener(this);
                c.add(l);
                //f.getContentPane().setBackground(Color.ORANGE);
                f.setVisible(true);
        }
        public void actionPerformed(ActionEvent ae){
                String s=new String(ae.getActionCommand());
                if((s).equals("Insert")){
```

```
                try{
                        t.setText("1 row Inserted "+t1.getText());
                        Class.forName("oracle.jdbc.OracleDriver");
                        Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","akshitha","vasavi");
                        //Statement stmt=con.createStatement();
                        //StringTokenizer st=new StringTokenizer(t1.getText(),",");
        //String data_type=st.nextToken();
                        //String syntax=st.nextToken();
                        //String description=st.nextToken();
                        //String sql = "select syntax, description from High_level_language
where data_type='" + t1.getText() + "'";
                        String sql = "select * from High_level_language where data_type='"
+ t1.getText() + "'";
                        PreparedStatement ps = con.prepareStatement(sql);
                        ResultSet rs = ps.executeQuery();
                        while(rs.next())
                        {
                                t4.append(rs.getString(1) + rs.getString(2) + rs.getString(3));
                        }
                        //stmt.executeUpdate("insert into
High_level_languagevalues('"+data_type+"','"+syntax+"','"+description+"')");
                        con.close();
                }
                catch (Exception e) {

                        System.out.println(e);
                        t.setText("Error Occured!!");

                }
                        t2.setText("");
                        //t3.setText("");
```

```
                t4.setText("");
            }
            else if((s).equals("Update")){
                try{
                    t.setText("1 row Updated "+t2.getText());
                    Class.forName("oracle.jdbc.OracleDriver");
                    Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","akshitha","vasavi");
                    Statement stmt=con.createStatement();
                    StringTokenizer st=new StringTokenizer(t2.getText(),",");
                    String data_type=st.nextToken();
                    String syntax=st.nextToken();
                    String description=st.nextToken();

                    stmt.executeUpdate("Update High_level_language set
data_type='"+data_type+"' where syntax='"+syntax+"' and description='"+description+"'");
                    con.close();
                }
                catch(Exception e){
                    System.out.println(e);
                    t.setText("Error Occured!!");
                }
                t1.setText("");
                //t3.setText("");
                t4.setText("");
            }
            /*else if((s).equals("Delete")){
                try{
                    t.setText("Deleted 1 row with pid "+t3.getText());
                    Class.forName("oracle.jdbc.OracleDriver");
                    Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","akshitha","vasavi");
```

```
                                Statement stmt=con.createStatement();
                                int p_id = Integer.parseInt(t3.getText());
                                stmt.executeUpdate("delete from High_level_language where
p_id="+p_id+"");
                              con.close();
                        }
                        catch(Exception e){
                                System.out.println(e);
                                t.setText("Error Occured!!");
                        }
                        t1.setText("");
                        t2.setText("");
                        t4.setText("");
            }*/
                /*else if((s).equals("Retrieve")){
                        try{
                                t.setText("Retrieved rows from Deficiency table");
                                Class.forName("oracle.jdbc.OracleDriver");
                                Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","lalitha","vasavi");
                                Statement stmt=con.createStatement();
                                ResultSet rs=stmt.executeQuery("select * from deficiency");
                                String str=new String();
                                while(rs.next())
                                    str=str+(rs.getString(1)+"  "+rs.getString(2)+"
"+rs.getInt(3)+"\n");
                                t4.setText(str);
                                con.close();
                        }
                        catch(Exception e){
                                t.setText("Error Occured!!");
                        }
```

```
                    t1.setText("");
                    t2.setText("");
                    t3.setText("");
                }*/
        }
        public static void main(String[] args){
                new High_level_language();
        }
}
```

TABLE : MEDIUM_LEVEL_LANGUAGE

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import java.util.*;
public class Medium_level_language implements ActionListener{
        private JFrame f=new JFrame("Medium_level_language");
        private JLabel l=new JLabel("");
        private JLabel l1=new JLabel("Enter datatype to insert");
        private JLabel l2=new JLabel("Enter datatype to update");
        //private JLabel l3=new JLabel("Enter datatype to delete");
        //private JLabel l4=new JLabel("Result");
        //private JLabel l5=new JLabel("DEFICIENCY");
        private JButton b1=new JButton("Insert");
        private JButton b2=new JButton("Update");
        //private JButton b3=new JButton("Delete");
        //private JButton b4=new JButton("Retrieve");
        private JTextField t1=new JTextField();
        private JTextField t2=new JTextField();
        //private JTextField t3=new JTextField();
        private JTextField t=new JTextField();
```

```
	private JTextArea t4=new JTextArea();
	private JScrollPane scrollBar=new
JScrollPane(t4,JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,JScrollPane.HORIZONTAL_SCROLLBAR_AL
WAYS);
	public Medium_level_language() {
		f.setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);
		f.setBounds(100,150,1000,400);
		Container c=f.getContentPane();
		f.getContentPane().add(l1);
		f.getContentPane().add(l2);
		//f.getContentPane().add(l3);
		//f.getContentPane().add(l4);
		//f.getContentPane().add(l5);
		f.getContentPane().add(b1);
		f.getContentPane().add(b2);
		//f.getContentPane().add(b3);
		//f.getContentPane().add(b4);
		f.getContentPane().add(t1);
		f.getContentPane().add(t2);
		//f.getContentPane().add(t3);
		f.getContentPane().add(t);
		f.getContentPane().add(scrollBar);
		scrollBar.setBounds(690,80,250,150);
		l.setBounds(20,30,50,50);
		l1.setBounds(60,80,250,30);
		l1.setOpaque(true);
		l1.setBackground(Color.WHITE);
		l2.setBounds(60,120,250,30);
		l2.setOpaque(true);
		l2.setBackground(Color.WHITE);
		//l3.setBounds(60,160,250,30);
		//l3.setOpaque(true);
```

```
                //l3.setBackground(Color.WHITE);
                //l4.setBounds(60,200,250,30);
                //l4.setOpaque(true);
                //l4.setBackground(Color.WHITE);
                //l5.setBounds(430,5,700,50);
//l5.setFont(new Font("Serif",Font.PLAIN,20));
                //l5.setForeground(Color.BLUE);
                b1.setBounds(570,80,100,30);
                b1.setFont(new Font("Times New Roman",Font.BOLD,17));
                b2.setBounds(570,120,100,30);
                b2.setFont(new Font("Times New Roman",Font.BOLD,17));
                //b3.setBounds(570,160,100,30);
                //b3.setFont(new Font("Times New Roman",Font.BOLD,17));
                //b4.setBounds(570,200,100,30);
                //b4.setFont(new Font("Times New Roman",Font.BOLD,17));
                t1.setBounds(330,80,220,30);
                t2.setBounds(330,120,220,30);
                //t3.setBounds(330,160,220,30);
                t.setBounds(330,200,220,30);
                t4.setEditable(false);
                b1.addActionListener(this);
                b2.addActionListener(this);
                //b3.addActionListener(this);
                //b4.addActionListener(this);
                c.add(l);
                //f.getContentPane().setBackground(Color.ORANGE);
                f.setVisible(true);
        }
        public void actionPerformed(ActionEvent ae){
                String s=new String(ae.getActionCommand());
                if((s).equals("Insert")){
```

```
                try{
                        t.setText("1 row Inserted "+t1.getText());
                        Class.forName("oracle.jdbc.OracleDriver");
                        Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","akshitha","vasavi");
                        /*Statement stmt=con.createStatement();
                        StringTokenizer st=new StringTokenizer(t1.getText(),",");
        String data_type=st.nextToken();
                        String syntax=st.nextToken();
                        String description=st.nextToken();
                        stmt.executeUpdate("insert into
Medium_level_languagevalues('"+data_type+"','"+syntax+"','"+description+"')");*/
                        String sql = "select * from Medium_level_language where
data_type='" + t1.getText() + "'";
                        PreparedStatement ps = con.prepareStatement(sql);
                        ResultSet rs = ps.executeQuery();
                        while(rs.next())
                        {
                                t4.append(rs.getString(1) + rs.getString(2) + rs.getString(3));
                        }
                        con.close();
                }
                catch (Exception e) {
                        t.setText("Error Occured!!");
                }
                        t2.setText("");
                        //t3.setText("");
                        t4.setText("");
                }
                else if((s).equals("Update")){
                        try{
                                t.setText("1 row Updated "+t2.getText());
```

```
                    Class.forName("oracle.jdbc.OracleDriver");
                    Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","akshitha","vasavi");
                    Statement stmt=con.createStatement();
                    StringTokenizer st=new StringTokenizer(t2.getText(),",");
                    String data_type=st.nextToken();
                    String syntax=st.nextToken();
                    String description=st.nextToken();

                    stmt.executeUpdate("Update Medium_level_language set
data_type='"+data_type+"' where syntax='"+syntax+"' and description='"+description+"'");
                    con.close();
                }
                catch(Exception e){
                    System.out.println(e);
                    t.setText("Error Occured!!");
                }
                t1.setText("");
                //t3.setText("");
                t4.setText("");
            }
            /*else if((s).equals("Delete")){
                try{
                    t.setText("Deleted 1 row with pid "+t3.getText());
                    Class.forName("oracle.jdbc.OracleDriver");
                    Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","akshitha","vasavi");
                    Statement stmt=con.createStatement();
                    int p_id = Integer.parseInt(t3.getText());
                    stmt.executeUpdate("delete from Medium_level_language where
p_id="+p_id+"");
                  con.close();
```

```
                        }
                        catch(Exception e){
                                System.out.println(e);
                                t.setText("Error Occured!!");
                        }
                        t1.setText("");
                        t2.setText("");
                        t4.setText("");
            }*/
                /*else if((s).equals("Retrieve")){
                        try{
                                t.setText("Retrieved rows from Deficiency table");
                                Class.forName("oracle.jdbc.OracleDriver");
                                Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","lalitha","vasavi");
                                Statement stmt=con.createStatement();
                                ResultSet rs=stmt.executeQuery("select * from deficiency");
                                String str=new String();
                                while(rs.next())
                                    str=str+(rs.getString(1)+"  "+rs.getString(2)+"
"+rs.getInt(3)+"\n");
                                t4.setText(str);
                                con.close();
                        }
                        catch(Exception e){
                                t.setText("Error Occured!!");
                        }
                        t1.setText("");
                        t2.setText("");
                        t3.setText("");
            }*/
          }
```

```
	public static void main(String[] args){
		new Medium_level_language();
	}
}
```

TABLE : LOW_LEVEL_LANGUAGE

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import java.util.*;
public class Low_level_language implements ActionListener{
	private JFrame f=new JFrame("Low_level_language");
	private JLabel l=new JLabel("");
	private JLabel l1=new JLabel("Enter datatype to insert");
	private JLabel l2=new JLabel("Enter datatype to update");
	//private JLabel l3=new JLabel("Enter datatype to delete");
	//private JLabel l4=new JLabel("Result");
	//private JLabel l5=new JLabel("DEFICIENCY");
	private JButton b1=new JButton("Insert");
	private JButton b2=new JButton("Update");
	//private JButton b3=new JButton("Delete");
	//private JButton b4=new JButton("Retrieve");
	private JTextField t1=new JTextField();
	private JTextField t2=new JTextField();
	//private JTextField t3=new JTextField();
	private JTextField t=new JTextField();
	private JTextArea t4=new JTextArea();
	private JScrollPane scrollBar=new
JScrollPane(t4,JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,JScrollPane.HORIZONTAL_SCROLLBAR_AL
WAYS);
	public Low_level_language() {
```

```
f.setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);
f.setBounds(100,150,1000,400);
Container c=f.getContentPane();
f.getContentPane().add(l1);
f.getContentPane().add(l2);
//f.getContentPane().add(l3);
//f.getContentPane().add(l4);
//f.getContentPane().add(l5);
f.getContentPane().add(b1);
f.getContentPane().add(b2);
//f.getContentPane().add(b3);
//f.getContentPane().add(b4);
f.getContentPane().add(t1);
f.getContentPane().add(t2);
//f.getContentPane().add(t3);
f.getContentPane().add(t);
f.getContentPane().add(scrollBar);
scrollBar.setBounds(690,80,250,150);
l.setBounds(20,30,50,50);
l1.setBounds(60,80,250,30);
l1.setOpaque(true);
l1.setBackground(Color.WHITE);
l2.setBounds(60,120,250,30);
l2.setOpaque(true);
l2.setBackground(Color.WHITE);
//l3.setBounds(60,160,250,30);
//l3.setOpaque(true);
//l3.setBackground(Color.WHITE);
//l4.setBounds(60,200,250,30);
//l4.setOpaque(true);
//l4.setBackground(Color.WHITE);
```

```
                //l5.setBounds(430,5,700,50);
        //l5.setFont(new Font("Serif",Font.PLAIN,20));
                //l5.setForeground(Color.BLUE);
                b1.setBounds(570,80,100,30);
                b1.setFont(new Font("Times New Roman",Font.BOLD,17));
                b2.setBounds(570,120,100,30);
                b2.setFont(new Font("Times New Roman",Font.BOLD,17));
                //b3.setBounds(570,160,100,30);
                //b3.setFont(new Font("Times New Roman",Font.BOLD,17));
                //b4.setBounds(570,200,100,30);
                //b4.setFont(new Font("Times New Roman",Font.BOLD,17));
                t1.setBounds(330,80,220,30);
                t2.setBounds(330,120,220,30);
                //t3.setBounds(330,160,220,30);
                t.setBounds(330,200,220,30);
                t4.setEditable(false);
                b1.addActionListener(this);
                b2.addActionListener(this);
                //b3.addActionListener(this);
                //b4.addActionListener(this);
                c.add(l);
                //f.getContentPane().setBackground(Color.ORANGE);
                f.setVisible(true);
        }
        public void actionPerformed(ActionEvent ae){
                String s=new String(ae.getActionCommand());
                if((s).equals("Insert")){
                        try{
                                t.setText("1 row Inserted "+t1.getText());
                                Class.forName("oracle.jdbc.OracleDriver");
```

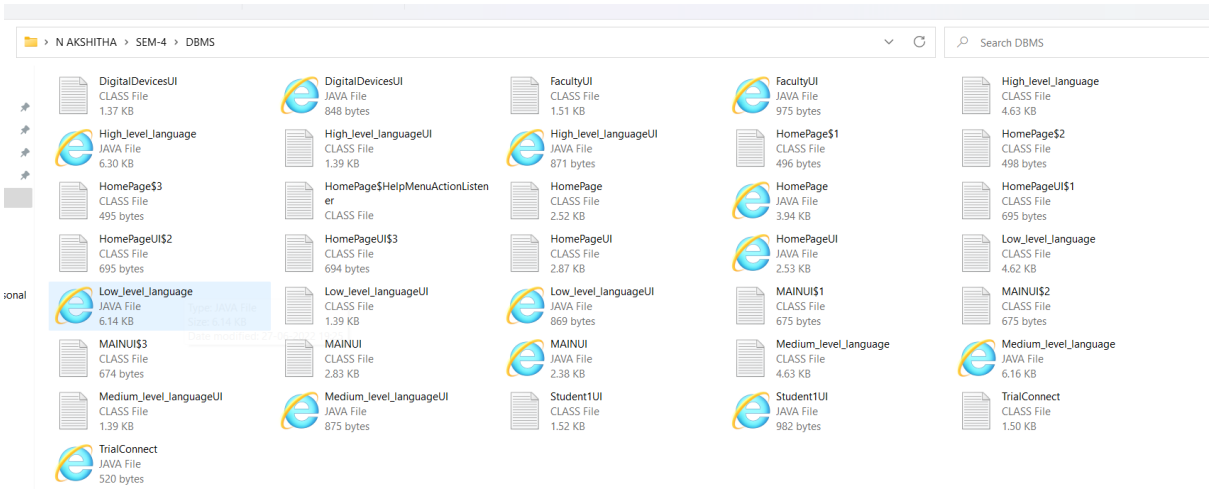
```
                                Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","akshitha","vasavi");
                                /*Statement stmt=con.createStatement();
                                StringTokenizer st=new StringTokenizer(t1.getText(),",");
        String data_type=st.nextToken();
                                String syntax=st.nextToken();
                                String description=st.nextToken();
                                stmt.executeUpdate("insert into
Low_level_languagevalues('"+data_type+"','"+syntax+"','"+description+"')");*/
                                String sql = "select * from Low_level_language where data_type='" +
t1.getText() + "'";
                                PreparedStatement ps = con.prepareStatement(sql);
                                ResultSet rs = ps.executeQuery();
                                while(rs.next())
                                {
                                        t4.append(rs.getString(1) + rs.getString(2) + rs.getString(3));
                                }
                                con.close();
                }
                catch (Exception e) {
                        t.setText("Error Occured!!");
                }
                        t2.setText("");
                        //t3.setText("");
                        t4.setText("");
                }
                else if((s).equals("Update")){
                        try{
                                t.setText("1 row Updated "+t2.getText());
                                Class.forName("oracle.jdbc.OracleDriver");
                                Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","akshitha","vasavi");
```

```
                    Statement stmt=con.createStatement();
                    StringTokenizer st=new StringTokenizer(t2.getText(),",");
                    String data_type=st.nextToken();
                    String syntax=st.nextToken();
                    String description=st.nextToken();

                    stmt.executeUpdate("Update Low_level_language set
data_type='"+data_type+"' where syntax='"+syntax+"' and description='"+description+"'");
                    con.close();
                }
                catch(Exception e){
                    System.out.println(e);
                    t.setText("Error Occured!!");
                }
                t1.setText("");
                //t3.setText("");
                t4.setText("");
            }
            /*else if((s).equals("Delete")){
                try{
                    t.setText("Deleted 1 row with pid "+t3.getText());
                    Class.forName("oracle.jdbc.OracleDriver");
                    Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","akshitha","vasavi");
                    Statement stmt=con.createStatement();
                    int p_id = Integer.parseInt(t3.getText());
                    stmt.executeUpdate("delete from Low_level_language where
p_id="+p_id+"");
                   con.close();
                }
                catch(Exception e){
                    System.out.println(e);
```

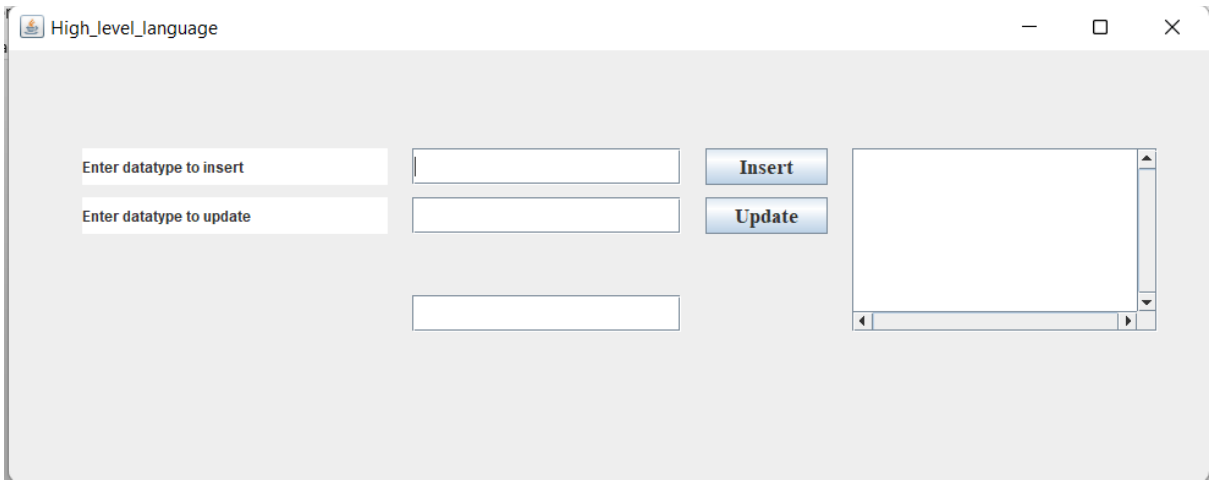
```
                    t.setText("Error Occured!!");
                }
                t1.setText("");
                t2.setText("");
                t4.setText("");
            }*/
        /*else if((s).equals("Retrieve")){
                try{
                    t.setText("Retrieved rows from Deficiency table");
                    Class.forName("oracle.jdbc.OracleDriver");
                    Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","lalitha","vasavi");
                    Statement stmt=con.createStatement();
                    ResultSet rs=stmt.executeQuery("select * from deficiency");
                    String str=new String();
                    while(rs.next())
                        str=str+(rs.getString(1)+"  "+rs.getString(2)+"
"+rs.getInt(3)+"\n");
                    t4.setText(str);
                    con.close();
                }
                catch(Exception e){
                    t.setText("Error Occured!!");
                }
                t1.setText("");
                t2.setText("");
                t3.setText("");
            }*/
        }
        public static void main(String[] args){
            new Low_level_language();
        }
```

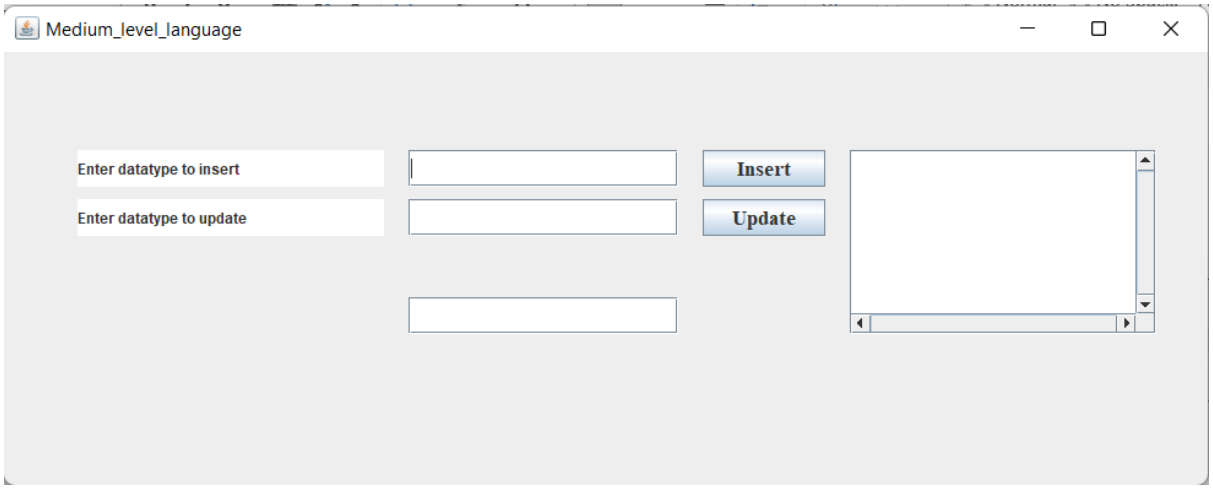
}

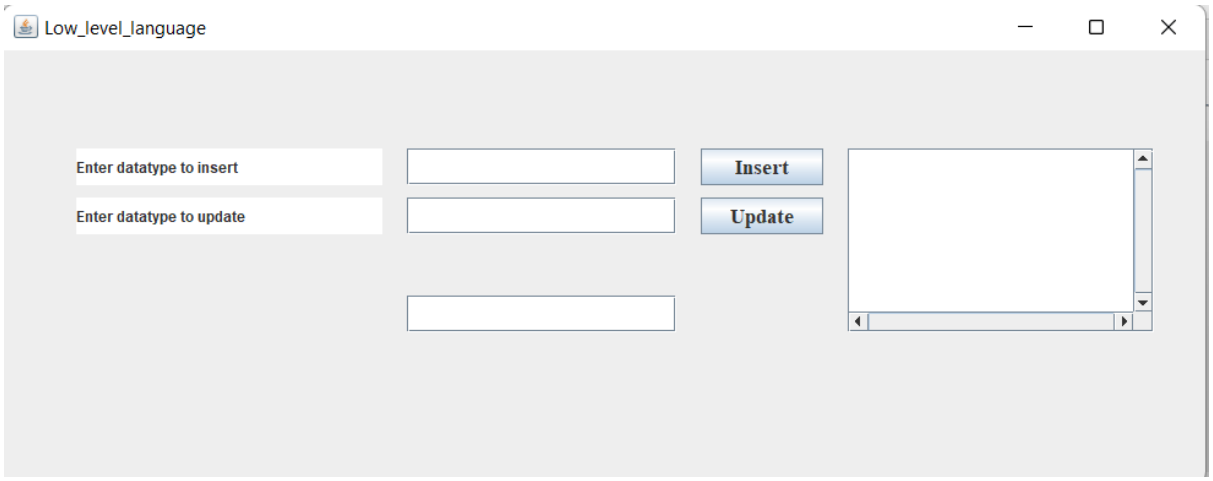
FOLDER STRUCTURE :

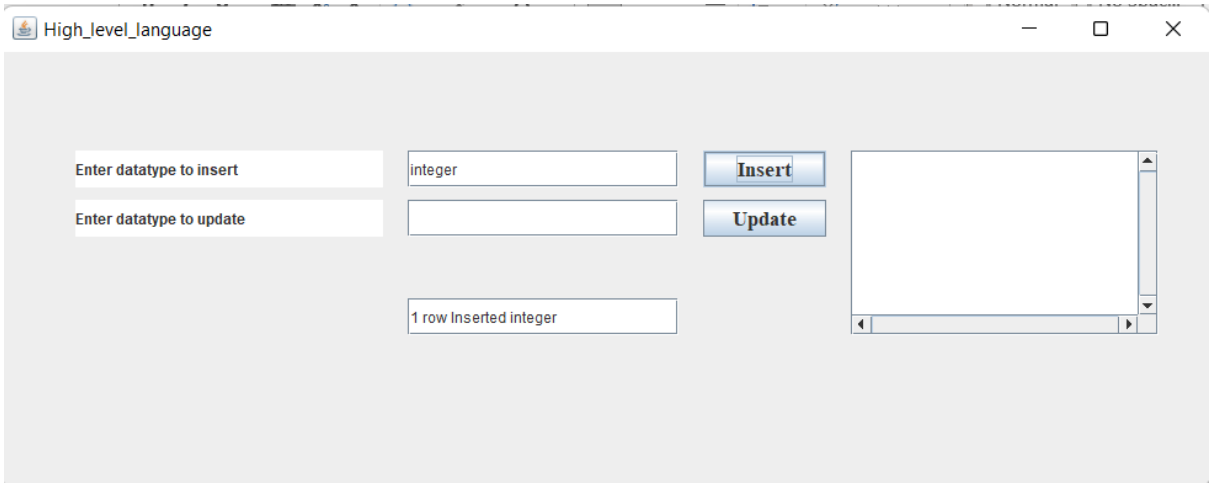


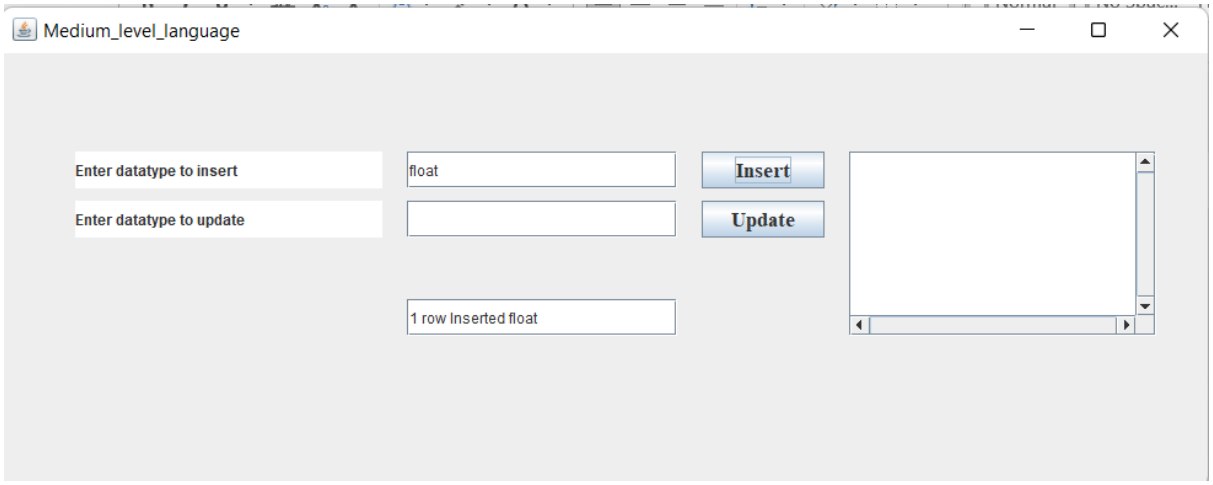
TESTING :

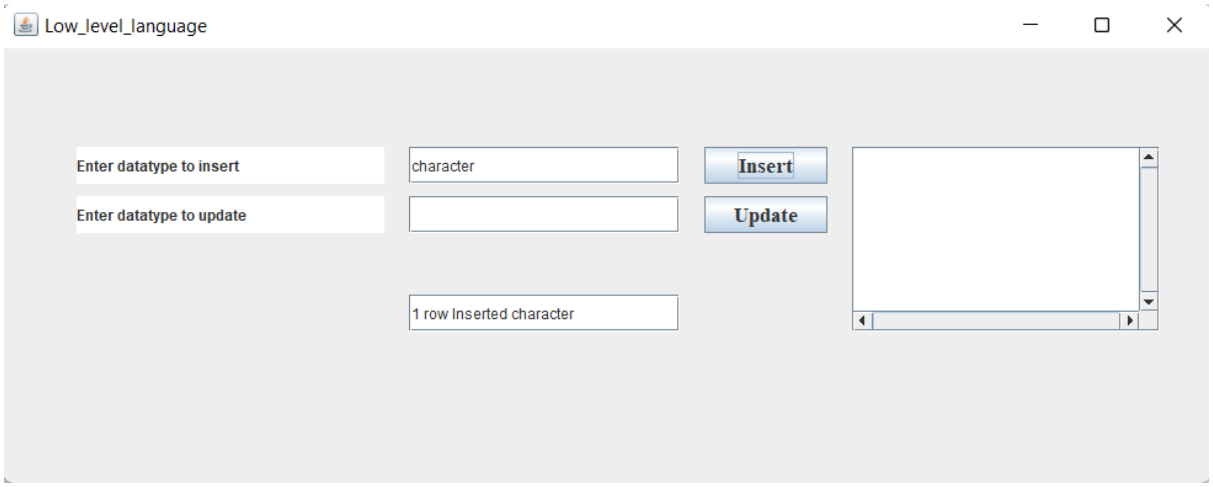



Medium_level_language
Enter datatype to insert
Enter datatype to update
Insert
Update


Low_level_language
Enter datatype to insert
Enter datatype to update
Insert
Update


High_level_language
Enter datatype to insert
Enter datatype to update
integer
1 row Inserted integer
Insert
Update





RESULT :

I had successfully completed PROJECT on “Syntax and Description in 3PL’S”.

DISCUSSION AND FUTURE WORK:

Data types and their syntaxes in 3 programming languages aims for the description of the datatypes of different programming languages when selected any datatype it should display the description and syntax of the database. That means here when we select any programming language it should display the respective language and the data types and their syntaxes and the required description.

So that when ever any one need any syntax and its description they can use this.

CONCLUSION:

Thus, a Java SWING based ***SYNTAX AND DESCRIPTION IN 3PL'S*** is created which is connected to the Oracle 11g database. Therefore, all the entries and details are directly updated on their respective tables created in the database.