Title:SUGGESTION BASED IMMUNE VACCINATION MANAGEMENT SYSTEM.

JAVA Suggestion Based Immune Vaccination 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.Varun Kumar<1602 18737115>

Department of Information Technology

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31

Title:SUGGESTION BASED IMMUNE VACCINATION MANAGEMENT SYSTEM.

AIM:

To suggest a proper vaccine management for proper immunity and growth for children.

ABSTRACT:

Suggestion based immune vaccination management system is basically about vaccines that should be taken and that are already taken by people before. It stores data(Registration ID,Name,Gender,DateOfBirth,HealthIssues), it stores data ,who took which vaccine and when did they take, it tells the maximum possible age for all the vaccines that should be taken,it tells which vaccine is to be taken to prevent disease or a problem. So, in a nutshell, this project provides an insight into how suggestion based vaccination management can be implemented efficiently. It gives a detailed information about which vaccine is the best fit to prevent any life-threatening diseases.

SOFTWARE USED:

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

REQUIREMENT ANALYSIS:

List of tables:

Candidate

Vaccine

Candidate_Vaccine

List of attributes with their domain types:

Candidate:

Title:SUGGESTION BASED IMMUNE VACCINATION MANAGEMENT SYSTEM.

Registration Number:Registration_Number-Number()

gender:gender-varchar2()

name:name-varchar2()

Health issues at time of birth-varchar2()

Date of birth-date()

Vaccine:

Vaccinename:varchar2()

Maxagetogetvaccine-number()

Genderspecification-Varchar2()

Dosage-Number()

Toprevent-Varchar2()

Candidate_Vaccine:

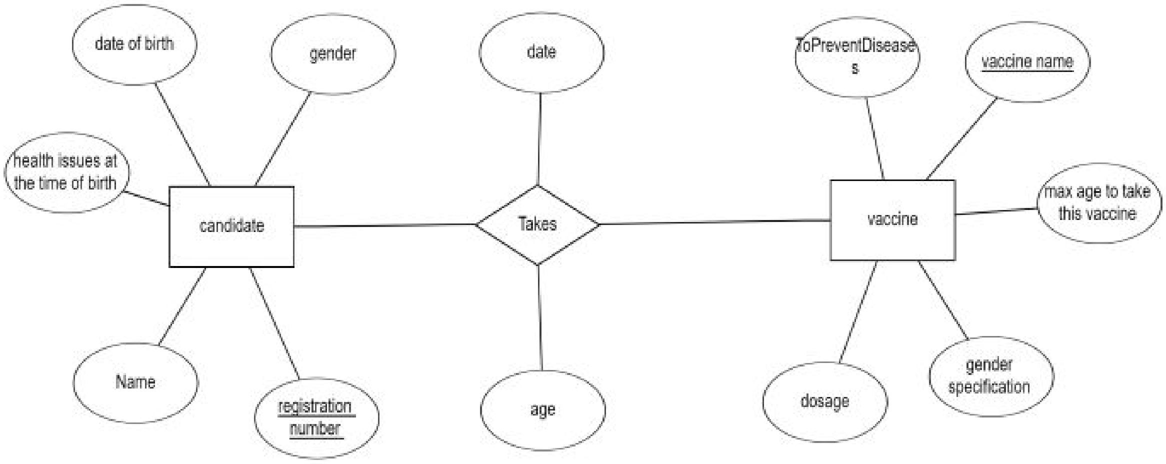
Registration_id:Number()

Vaccinename:varchar2()

Day:date()

Ageatthetimeofvaccination:Number()

ER DIAGRAM:



SQL:

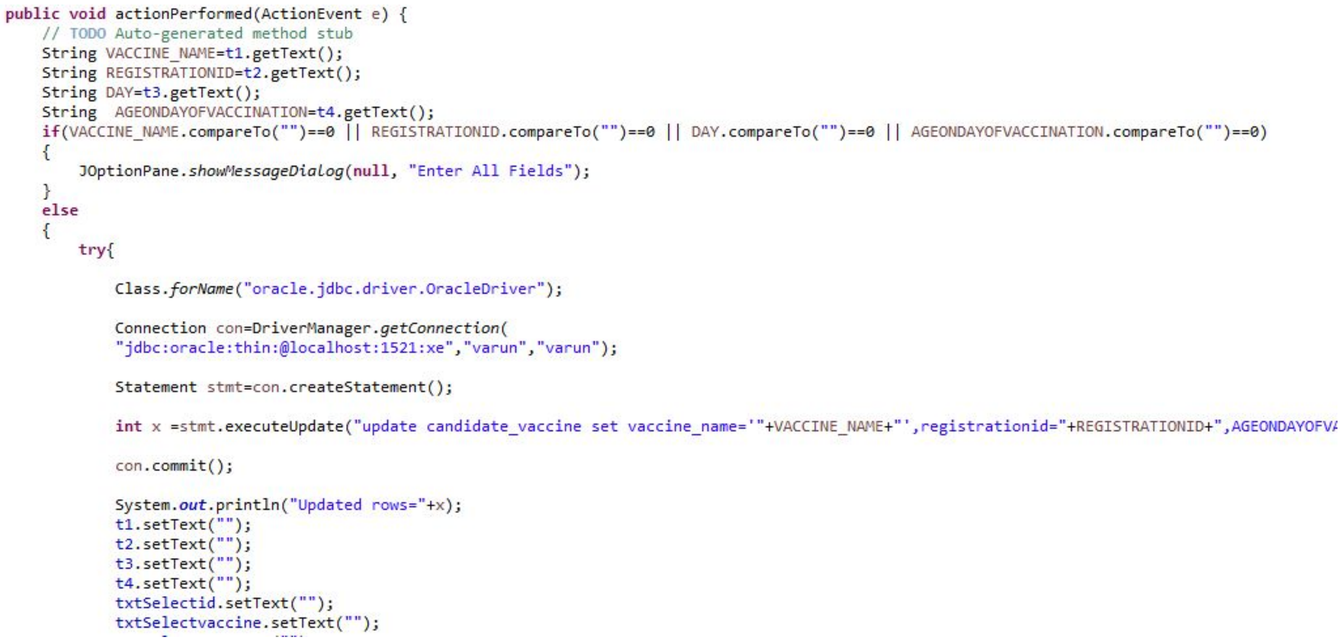
Structure Query Language(SQL) is a database query language used for storing and managing data in Relational DBMS. SQL was the first commercial language introduced for E.F Codd's **Relational** model of database. Today almost all RDBMS (MySql, Oracle, Infomix, Sybase, MS Access) use **SQL** as the standard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

Java-SQL Connectivity using JDBC:

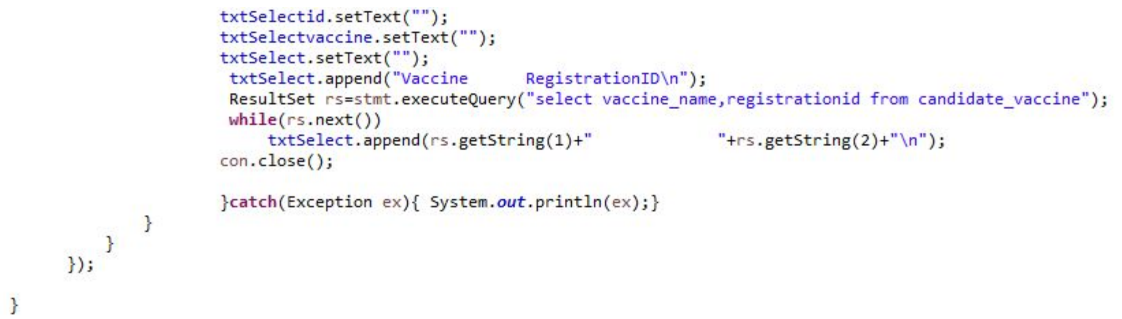
Java Database Connectivity (**JDBC**) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from 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:

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



```
public void actionPerformed(ActionEvent e) {
    // TODO Auto-generated method stub
    String VACCINE_NAME=t1.getText();
    String REGISTRATIONID=t2.getText();
    String DAY=t3.getText();
    String  AGEONDAYOFVACCINATION=t4.getText();
    if(VACCINE_NAME.compareTo("")==0 || REGISTRATIONID.compareTo("")==0 || DAY.compareTo("")==0 || AGEONDAYOFVACCINATION.compareTo("")==0)
    {
        JOptionPane.showMessageDialog(null, "Enter All Fields");
    }
    else
    {
        try{

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

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

            Statement stmt=con.createStatement();

            int x =stmt.executeUpdate("update candidate_vaccine set vaccine_name='"+VACCINE_NAME+"',registrationid="+REGISTRATIONID+",AGEONDAYOFVA

            con.commit();

            System.out.println("Updated rows="+x);
            t1.setText("");
            t2.setText("");
            t3.setText("");
            t4.setText("");
            txtSelectid.setText("");
            txtSelectvaccine.setText("");
```



```
                        txtSelectid.setText("");
                        txtSelectvaccine.setText("");
                        txtSelect.setText("");
                         txtSelect.append("Vaccine      RegistrationID\n");
                         ResultSet rs=stmt.executeQuery("select vaccine_name,registrationid from candidate_vaccine");
                         while(rs.next())
                             txtSelect.append(rs.getString(1)+"             "+rs.getString(2)+"\n");
                        con.close();

                        }catch(Exception ex){ System.out.println(ex);}
                }
            }
        });

}
```

Table Created in SQL for above mentioned purpose is as:

```
create table register( stid number(5),sname varchar(20),ph number(10),dept varchar2(20),city varchar2(20),mail varchar2(30),fname varchar(15),addr varchar2(20));
```

Program:

CandidateInsert program

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
public class candidateinsert extends JFrame {
                private JPanel panel;
                private JLabel label;
                private JTextField t1;
                private JLabel label1;
```

```
        private JTextField t2;
        private JLabel label2;
        private JTextField t3;
        private JTextField t4;
        private JLabel label3;
        private JLabel label4;
        private JTextField t5;
        private JButton insert;
        private JMenuItem menuinsert;
    private JMenuItem delete;
    private JMenuItem view;
    private JMenuItem insert1;
    private JMenuItem delete1;
    private JMenuItem view1;
    private JMenuItem insert2;
    private JMenuItem delete2;
    private JMenuItem view2;
    private JFrame frame;
    private JMenuBar menubar;
```

```
private JMenu candidate;
private JMenu vaccine;
private JMenu data;
    public candidateinsert()
    {
     panel=new JPanel(new FlowLayout());
     label=new JLabel("candidatename");
     t1=new JTextField(20);
     label2=new JLabel("registration number");
     label1=new JLabel("dob");
     label3=new JLabel("gender");
     label4=new JLabel("Healthisssuesatbirth");
     insert=new JButton("insert");
     t2=new JTextField(10);
     t3=new JTextField(10);
     t4=new JTextField(15);
     t5=new JTextField(30);
     candidate=new JMenu("candidate");
          vaccine=new JMenu("vaccine");
```

```
                   data=new JMenu("data");
                   menuinsert=new JMenuItem("insert");
                  delete=new JMenuItem("delete");
                  view=new JMenuItem("view");
                  insert1=new JMenuItem("insert");
                 delete1=new JMenuItem("delete");
                  view1=new JMenuItem("view");
                 insert2=new JMenuItem("insert");
                  delete2=new JMenuItem("delete");
                  view2=new JMenuItem("view");
               frame=new JFrame("Menu");
               menubar=new JMenuBar();
         this.add(panel);


        this.setVisible(true);
        this.setDefaultCloseOperation(3);
        this.setSize(1000,300);


this.setJMenuBar(menubar);
```

```
panel.add(label);
panel.add(t1);
panel.add(label2);
panel.add(t2);
panel.add(label1);
panel.add(t3);
panel.add(label3);
panel.add(t4);
panel.add(label4);
panel.add(t5);
panel.add(insert);
```

```
  menubar.add(candidate);


  candidate.add(menuinsert);

  candidate.add(delete);

  candidate.add(view);


  menubar.add(vaccine);

  vaccine.add(insert1);

 vaccine.add(delete1);

 vaccine.add(view1);

  menubar.add(data);

  data.add(insert2);

data.add(delete2);

data.add(view2);

menuinsert.addActionListener(new ActionListener() {


                @Override

                public void actionPerformed(ActionEvent e) {

                    new candidateinsert();
```

```
                  dispose();


              }

 }

 );

 view.addActionListener(new ActionListener() {


              @Override

              public void actionPerformed(ActionEvent e) {


                 new candidateview();

                 dispose();

              }

          }




);

 delete.addActionListener(new ActionListener() {
```

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

                                new candidatedelete();
                                dispose();
                          }
                  }



);
 insert1.addActionListener(new ActionListener() {

                          @Override
                          public void actionPerformed(ActionEvent e) {

                                new vaccineinsert();
                                dispose();
```

```
                    }
                }




        );
         delete1.addActionListener(new ActionListener() {

                    @Override
                    public void actionPerformed(ActionEvent e) {

                        new vaccinedelete();
                        dispose();
                    }
                }




        );
```

```
view1.addActionListener(new ActionListener() {

                @Override
                public void actionPerformed(ActionEvent e) {



                    new vaccineview();
                    dispose();
                }
            }




);
 insert2.addActionListener(new ActionListener() {

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

```
                                new datainsert();

                                dispose();

                            }

                    }





          );

           delete2.addActionListener(new ActionListener() {


                            @Override

                            public void actionPerformed(ActionEvent e) {


                                new datadelete();

                                dispose();

                            }

                    }
```

```
);
 view2.addActionListener(new ActionListener() {

                @Override
                public void actionPerformed(ActionEvent e) {

                    new dataview();
                    dispose();
                }
           }



);

 insert.addActionListener(new ActionListener() {

                @Override
                public void actionPerformed(ActionEvent e) {
                    // TODO Auto-generated method stub
                    if(t1.getText().compareTo("")==0 ||
t2.getText().compareTo("")==0 || t3.getText().compareTo("")==0 ||
t4.getText().compareTo("")==0 || t5.getText().compareTo("")==0)
                    {

JOptionPane.showMessageDialog(null, "Enter All Fields");
                    }
                    else
                    {
                        try{


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


                            Connection
con=DriverManager.getConnection(

"jdbc:oracle:thin:@localhost:1521:xe","varun","varun");
```

```
                                        Statement
stmt=con.createStatement();
                                        int
x=stmt.executeUpdate("insert into candidate
values("+t2.getText()+",'"+t1.getText()+"','"+t3.getText()+"','"+t4.getTe
xt()+"','"+t5.getText()+"')");
                                        System.out.println("Insert
rows="+x);
                                        con.commit();
                                        t1.setText("");
                                        t2.setText("");
                                        t3.setText("");
                                        t4.setText("");
                                        t5.setText("");
                                        con.close();
                                        }catch(Exception ex){
System.out.println(ex);}
                              }
                         }
                    });
          }
```

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


		}


}
```

Candidate Delete:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
public class candidatedelete extends JFrame {
		private JPanel panel;
```

```
        private JLabel label;
        private JTextField t1;
        private JLabel label1;
        private JTextField t2;
        private JLabel label2;
        private JTextField t3;
        private JTextField t4;
        private JLabel label3;
        private JLabel label4;
        private JTextField t5;
        private JButton delete;
        private JMenuItem menuinsert;
    private JMenuItem menudelete;
    private JMenuItem view;
    private JMenuItem insert1;
    private JMenuItem delete1;
    private JMenuItem view1;
    private JMenuItem insert2;
    private JMenuItem delete2;
```

```
private JMenuItem view2;

private JFrame frame;

private JMenuBar menubar;

private JMenu candidate;

private JMenu vaccine;

private JMenu data;

private JTextArea txtSelect;

private JLabel lblselect;

private JTextField txtSelectPrimary;

private JButton btnview;


        public candidatedelete()


        {

         txtSelect=new JTextArea(10,20);

         lblselect=new JLabel("Select ");

         txtSelectPrimary=new JTextField(20);

         btnview=new JButton("view");

       delete=new JButton("delete");
```

```
panel=new JPanel(new FlowLayout());
label=new JLabel("candidatename");
t1=new JTextField(20);
label2=new JLabel("registration number");
label1=new JLabel("dob");
label3=new JLabel("gender");
label4=new JLabel("Healthisssuesatbirth");


t2=new JTextField(10);
t3=new JTextField(10);
t4=new JTextField(15);
t5=new JTextField(30);
candidate=new JMenu("candidate");
          vaccine=new JMenu("vaccine");
          data=new JMenu("data");
          menuinsert=new JMenuItem("insert");
         menudelete=new JMenuItem("delete");
         view=new JMenuItem("view");
```

```
                  insert1=new JMenuItem("insert");
                 delete1=new JMenuItem("delete");
                  view1=new JMenuItem("view");
                 insert2=new JMenuItem("insert");
                  delete2=new JMenuItem("delete");
                  view2=new JMenuItem("view");
               frame=new JFrame("Menu");
               menubar=new JMenuBar();
         this.add(panel);


        this.setVisible(true);
        this.setDefaultCloseOperation(3);
        this.setSize(1000,300);


this.setJMenuBar(menubar);
     panel.add(txtSelect);
     panel.add(lblselect);
     panel.add(txtSelectPrimary);
     panel.add(btnview);
        panel.add(label);
        panel.add(t1);
        panel.add(label2);
        panel.add(t2);
        panel.add(label1);
        panel.add(t3);
        panel.add(label3);
        panel.add(t4);
        panel.add(label4);
        panel.add(t5);
        panel.add(delete);








        menubar.add(candidate);
```

```
  candidate.add(menuinsert);
  candidate.add(menudelete);
  candidate.add(view);

  menubar.add(vaccine);
  vaccine.add(insert1);
 vaccine.add(delete1);
 vaccine.add(view1);
  menubar.add(data);
  data.add(insert2);
data.add(delete2);
data.add(view2);
menuinsert.addActionListener(new ActionListener() {

                    @Override
                    public void actionPerformed(ActionEvent e) {
                        new candidateinsert();
                        dispose();
```

```
              }
 }
 );
 view.addActionListener(new ActionListener() {

              @Override
              public void actionPerformed(ActionEvent e) {

                  new candidateview();
                  dispose();
              }
          }




);
 delete.addActionListener(new ActionListener() {
```

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

                        new candidatedelete();
                        dispose();
                    }
                }



        );
         insert1.addActionListener(new ActionListener() {

                    @Override
                    public void actionPerformed(ActionEvent e) {

                        new vaccineinsert();
                        dispose();
                    }
                }




        );
         delete1.addActionListener(new ActionListener() {

                    @Override
                    public void actionPerformed(ActionEvent e) {

                        new vaccinedelete();
                        dispose();
                    }
                }




        );
         view1.addActionListener(new ActionListener() {
```

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



                       new vaccineview();
                       dispose();
                    }
                }




        );
         insert2.addActionListener(new ActionListener() {

                    @Override
                    public void actionPerformed(ActionEvent e) {

                       new datainsert();
                    dispose();
                }
           }




);
 delete2.addActionListener(new ActionListener() {

                @Override
                public void actionPerformed(ActionEvent e) {

                    new datadelete();
                    dispose();
                }
           }
```

```
);
 view2.addActionListener(new ActionListener() {

                 @Override
                 public void actionPerformed(ActionEvent e) {

                     new dataview();
                     dispose();
                 }
             }




);

 try{

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

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


    Statement stmt=con.createStatement();


    ResultSet rs=stmt.executeQuery("select * from candidate");
    while(rs.next())
        txtSelect.append(rs.getInt(1)+"\n");




    con.close();


    }catch(Exception e){ System.out.println(e);}
   btnview.addActionListener(new ActionListener() {


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

```
if(txtSelectPrimary.getText().compareTo("")==0)
                                        {

JOptionPane.showMessageDialog(null, "Enter Registartion Number");
                                        }
                                        else
                                        {
                                              try{



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


                                                     Connection
con=DriverManager.getConnection(


"jdbc:oracle:thin:@localhost:1521:xe","varun","varun");


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

```
                                                    ResultSet
rs=stmt.executeQuery("select * from candidate where
REGISTRATIONID="+txtSelectPrimary.getText()+"");
                                                    while(rs.next())
                                                    {

t2.setText(rs.getInt(1)+"");

t1.setText(rs.getString(2)+"");

t3.setText(rs.getString(3)+"");

t4.setText(rs.getString(4)+"");

t5.setText(rs.getString(5));
                                                    }

                                                    con.close();

                                                    }catch(Exception ex){
System.out.println(ex);}
```

```
                                        }
                                }
                        });
        delete.addActionListener(new ActionListener() {


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

if(txtSelectPrimary.getText().compareTo("")==0)
                                        {

JOptionPane.showMessageDialog(null, "Select Registartion number");
                                        }
                                        else
                                        {
                                                try
                                                {

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

```
                                   Connection
con=DriverManager.getConnection(

"jdbc:oracle:thin:@localhost:1521:xe","varun","varun");

                                   Statement
stmt=con.createStatement();

                                   int x=stmt.executeUpdate("delete
from candidate where
REGISTRATIONID="+txtSelectPrimary.getText()+"");
                                   System.out.println("Total Updated
Rows "+x);
                                   con.commit();

                                   txtSelect.setText("");
                                   txtSelectPrimary.setText("");
                                   t1.setText("");
                                   t2.setText("");
                                   t3.setText("");
```

```
                                                t4.setText("");
                                                t5.setText("");
                         ResultSet rs=stmt.executeQuery("select * from
candidate");
                         while(rs.next())
                              txtSelect.append(rs.getInt(1)+"\n");
                                                con.close();


                                                }catch(Exception ex){
System.out.println(ex);}
                                          }
                                    }
                         });
            }
            public static void main(String args[])
            {
             new candidatedelete();


            }
```

```
}
```

Candidate view:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
public class candidateview extends JFrame {
        private JPanel panel;
        private JLabel label;
        private JTextField t1;
        private JLabel label1;
        private JTextField t2;
        private JLabel label2;
        private JTextField t3;
        private JTextField t4;
        private JLabel label3;
        private JLabel label4;
        private JTextField t5;
        private JButton view;
        private JMenuItem menuinsert;
    private JMenuItem menudelete;
    private JMenuItem menuview;
    private JMenuItem insert1;
    private JMenuItem delete1;
    private JMenuItem view1;
    private JMenuItem insert2;
    private JMenuItem delete2;
    private JMenuItem view2;
    private JFrame frame;
    private JMenuBar menubar;
    private JMenu candidate;
    private JMenu vaccine;
    private JMenu data;
    private JTextArea txtSelect;
```

```
private JLabel lblselect;

private JTextField txtSelectPrimary;

private JButton btnview;


        public candidateview()


        {

         txtSelect=new JTextArea(10,20);

         lblselect=new JLabel("Select ");

         txtSelectPrimary=new JTextField(20);

         btnview=new JButton("view");

       view=new JButton("update");


         panel=new JPanel(new FlowLayout());

         label=new JLabel("candidatename");

         t1=new JTextField(20);

         label2=new JLabel("registration number");

         label1=new JLabel("DOB");

         label3=new JLabel("gender");
```

```
label4=new JLabel("Healthisssuesatbirth");


t2=new JTextField(10);

t3=new JTextField(10);

t4=new JTextField(15);

t5=new JTextField(30);

candidate=new JMenu("candidate");

     vaccine=new JMenu("vaccine");

     data=new JMenu("data");

     menuinsert=new JMenuItem("insert");

    menudelete=new JMenuItem("delete");

    menuview=new JMenuItem("view");

    insert1=new JMenuItem("insert");

    delete1=new JMenuItem("delete");

    view1=new JMenuItem("view");

    insert2=new JMenuItem("insert");

    delete2=new JMenuItem("delete");

    view2=new JMenuItem("view");

   frame=new JFrame("Menu");
```

```
               menubar=new JMenuBar();
         this.add(panel);


        this.setVisible(true);
        this.setDefaultCloseOperation(3);
        this.setSize(1000,300);


this.setJMenuBar(menubar);
     panel.add(txtSelect);
     panel.add(lblselect);
     panel.add(txtSelectPrimary);
     panel.add(btnview);


        panel.add(label);
        panel.add(t1);
        panel.add(label2);
        panel.add(t2);
        panel.add(label1);
        panel.add(t3);
        panel.add(label3);
        panel.add(t4);
        panel.add(label4);
        panel.add(t5);
        panel.add(view);










        menubar.add(candidate);


        candidate.add(menuinsert);
        candidate.add(menudelete);
        candidate.add(menuview);


        menubar.add(vaccine);
        vaccine.add(insert1);
```

```
 vaccine.add(delete1);
 vaccine.add(view1);
  menubar.add(data);
  data.add(insert2);
data.add(delete2);
data.add(view2);
menuinsert.addActionListener(new ActionListener() {

                @Override
                public void actionPerformed(ActionEvent e) {
                    new candidateinsert();
                    dispose();

                }
}
);
menuview.addActionListener(new ActionListener() {

                @Override
```

```
                    public void actionPerformed(ActionEvent e) {

                        new candidateview();
                        dispose();
                    }
                }



        );
         menudelete.addActionListener(new ActionListener() {

                    @Override
                    public void actionPerformed(ActionEvent e) {

                        new candidatedelete();
                        dispose();
                    }
                }
```

```
);
 insert1.addActionListener(new ActionListener() {

                    @Override
                    public void actionPerformed(ActionEvent e) {

                        new vaccineinsert();
                        dispose();
                    }
                }

);
 delete1.addActionListener(new ActionListener() {
```

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

                          new vaccinedelete();
                          dispose();
                    }
                }




      );
       view1.addActionListener(new ActionListener() {

                    @Override
                    public void actionPerformed(ActionEvent e) {



                          new vaccineview();
                          dispose();
                }
            }




    );
     insert2.addActionListener(new ActionListener() {

                @Override
                public void actionPerformed(ActionEvent e) {

                    new datainsert();
                    dispose();
                }
            }




    );
```

```
 delete2.addActionListener(new ActionListener() {


                  @Override
                  public void actionPerformed(ActionEvent e) {


                        new datadelete();
                        dispose();
                  }
            }




);
 view2.addActionListener(new ActionListener() {


                  @Override
                  public void actionPerformed(ActionEvent e) {


                        new dataview();
```

```
                                dispose();
                            }
                        }




             );

              try{

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

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

                Statement stmt=con.createStatement();

                ResultSet rs=stmt.executeQuery("select * from candidate");
                while(rs.next())
```

```
                    txtSelect.append(rs.getInt(1)+"\n");



                con.close();


                }catch(Exception e){ System.out.println(e);}
               btnview.addActionListener(new ActionListener() {

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

if(txtSelectPrimary.getText().compareTo("")==0)
                                   {

JOptionPane.showMessageDialog(null, "Enter Registartion Number");
                                   }
                                   else
                                   {
                                        try{
```

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

                                        Connection
con=DriverManager.getConnection(

"jdbc:oracle:thin:@localhost:1521:xe","varun","varun");

                                        Statement
stmt=con.createStatement();

                                        ResultSet
rs=stmt.executeQuery("select * from candidate where
REGISTRATIONID="+txtSelectPrimary.getText()+"");
                                        while(rs.next())
                                        {

t2.setText(rs.getInt(1)+"");

t1.setText(rs.getString(2)+"");
t3.setText(rs.getString(3)+"");

t4.setText(rs.getString(4)+"");

t5.setText(rs.getString(5));
                                }



                                con.close();


                                }catch(Exception ex){
System.out.println(ex);}
                        }
                    }
                });
       view.addActionListener(new ActionListener() {


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

```
                                        // TODO Auto-generated method stub
                                        if(t1.getText().compareTo("")==0 ||
t2.getText().compareTo("")==0 || t3.getText().compareTo("")==0 ||
t4.getText().compareTo("")==0 || t5.getText().compareTo("")==0)
                                        {

JOptionPane.showMessageDialog(null, "Enter All Fields");
                                        }
                                        else
                                        {
                                             try{


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


                                                       Connection
con=DriverManager.getConnection(

"jdbc:oracle:thin:@localhost:1521:xe","varun","varun");
```

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


                                        int
x=stmt.executeUpdate("update candidate set
REGISTRATIONID="+t2.getText()+",NAME='"+t1.getText()+"',GEND
ER='"+t4.getText()+"',HEALTHISSUES='"+t5.getText()+"' where
REGISTRATIONID="+txtSelect.getText()+"");
                                        System.out.println("Total
Updated Rows "+x);
                                        con.commit();


                                        txtSelect.setText("");
                                        txtSelectPrimary.setText("");
                                        t1.setText("");
                                        t2.setText("");
                                        t3.setText("");
                                        t4.setText("");
                                        t5.setText("");
                            ResultSet rs=stmt.executeQuery("select *
from candidate");
```

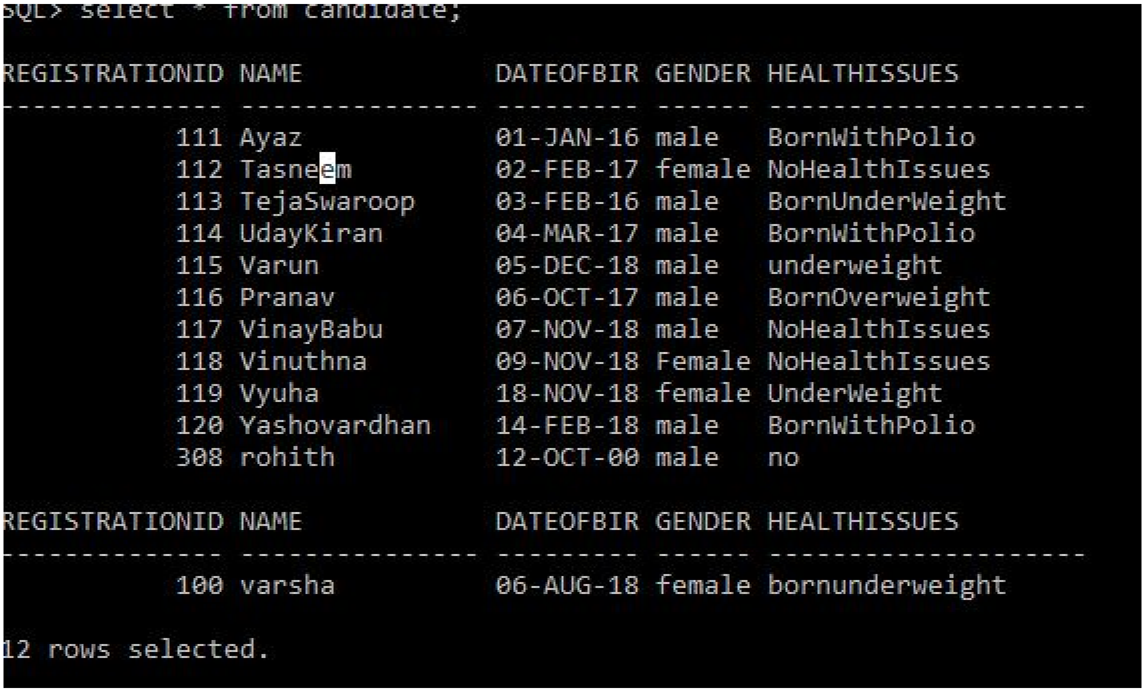
```
                    while(rs.next())
                        txtSelect.append(rs.getInt(1)+"\n");
                                con.close();


                                }catch(Exception ex){
System.out.println(ex);}
                        }
                    }
                });


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


        }


}
```

Title:SUGGESTION BASED IMMUNE VACCINATION MANAGEMENT SYSTEM.

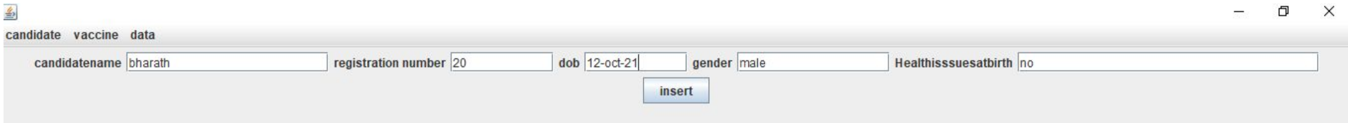
Title:SUGGESTION BASED IMMUNE VACCINATION MANAGEMENT SYSTEM.

OUTPUT SCREENSHOTS:

previous data in database

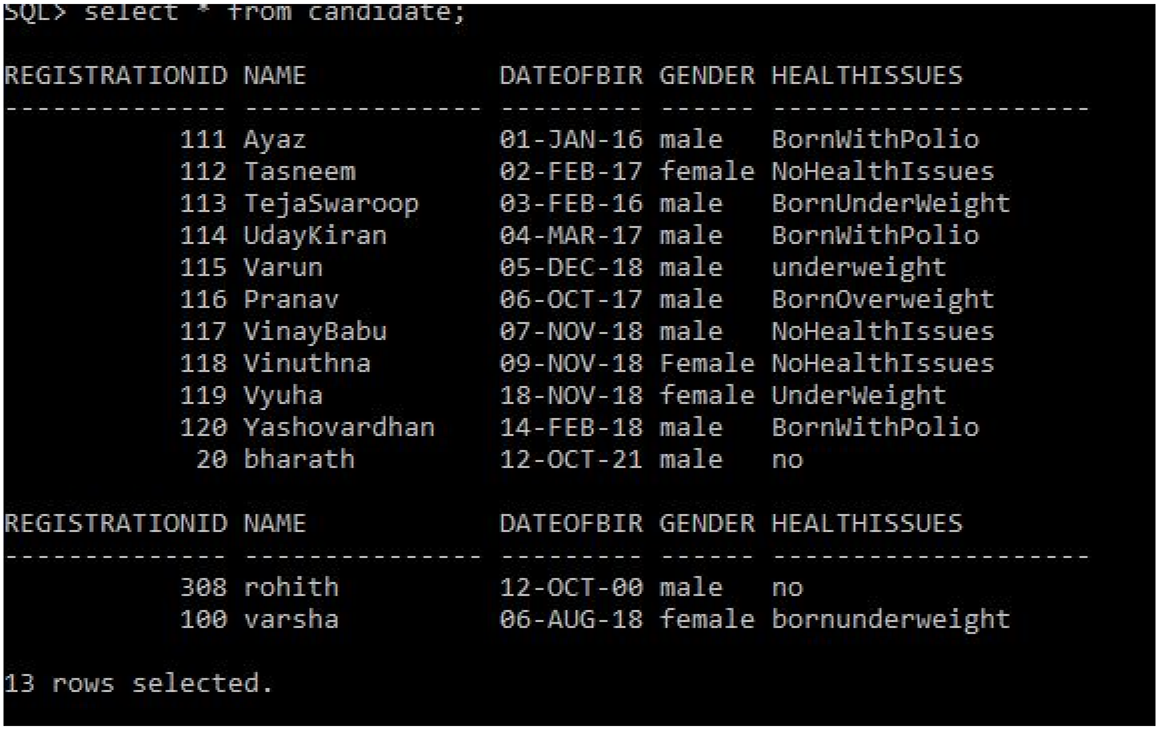


```
SQL> select * from candidate;

REGISTRATIONID NAME            DATEOFBIR GENDER HEALTHISSUES
-------------- --------------- --------- ------ --------------------
           111 Ayaz            01-JAN-16 male   BornWithPolio
           112 Tasneem         02-FEB-17 female NoHealthIssues
           113 TejaSwaroop     03-FEB-16 male   BornUnderWeight
           114 UdayKiran       04-MAR-17 male   BornWithPolio
           115 Varun           05-DEC-18 male   underweight
           116 Pranav          06-OCT-17 male   BornOverweight
           117 VinayBabu       07-NOV-18 male   NoHealthIssues
           118 Vinuthna        09-NOV-18 Female NoHealthIssues
           119 Vyuha           18-NOV-18 female UnderWeight
           120 Yashovardhan    14-FEB-18 male   BornWithPolio
           308 rohith          12-OCT-00 male   no

REGISTRATIONID NAME            DATEOFBIR GENDER HEALTHISSUES
-------------- --------------- --------- ------ --------------------
           100 varsha          06-AUG-18 female bornunderweight

12 rows selected.
```



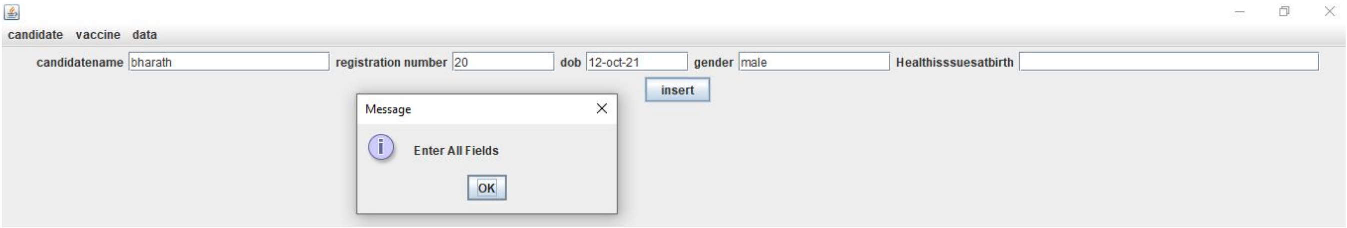
candidatename bharath registration number 20 dob 12-oct-21 gender male Healthisssuesatbirth no

Title:SUGGESTION BASED IMMUNE VACCINATION MANAGEMENT SYSTEM.



```
SQL> select * from candidate;

REGISTRATIONID NAME           DATEOFBIR GENDER HEALTHISSUES
-------------- -------------- --------- ------ --------------------
           111 Ayaz           01-JAN-16 male   BornWithPolio
           112 Tasneem        02-FEB-17 female NoHealthIssues
           113 TejaSwaroop    03-FEB-16 male   BornUnderWeight
           114 UdayKiran      04-MAR-17 male   BornWithPolio
           115 Varun          05-DEC-18 male   underweight
           116 Pranav         06-OCT-17 male   BornOverweight
           117 VinayBabu      07-NOV-18 male   NoHealthIssues
           118 Vinuthna       09-NOV-18 Female NoHealthIssues
           119 Vyuha          18-NOV-18 female UnderWeight
           120 Yashovardhan   14-FEB-18 male   BornWithPolio
            20 bharath        12-OCT-21 male   no

REGISTRATIONID NAME           DATEOFBIR GENDER HEALTHISSUES
-------------- -------------- --------- ------ --------------------
           308 rohith         12-OCT-00 male   no
           100 varsha         06-AUG-18 female bornunderweight

13 rows selected.
```

Error Message telling to enter all fields as per schema

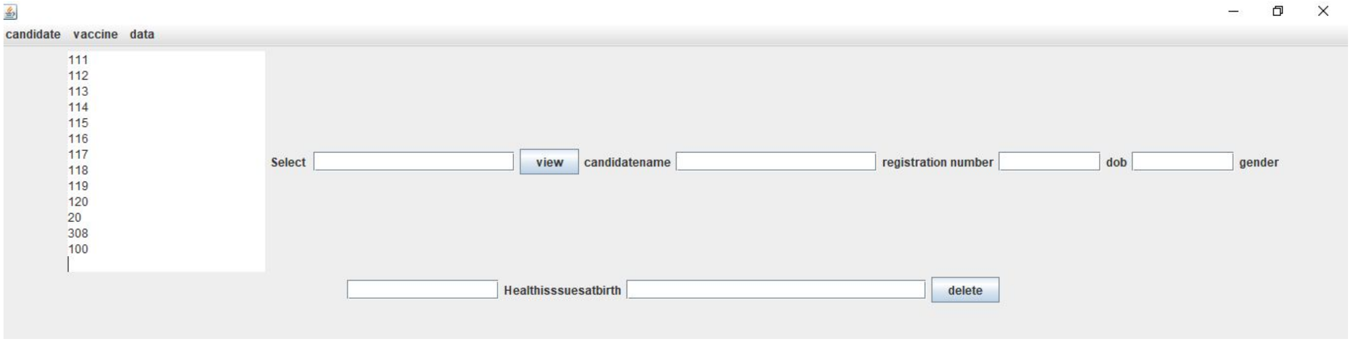


candidatename bharath registration number 20 dob 12-oct-21 gender male Healthisssuesatbirth

Message

Enter All Fields

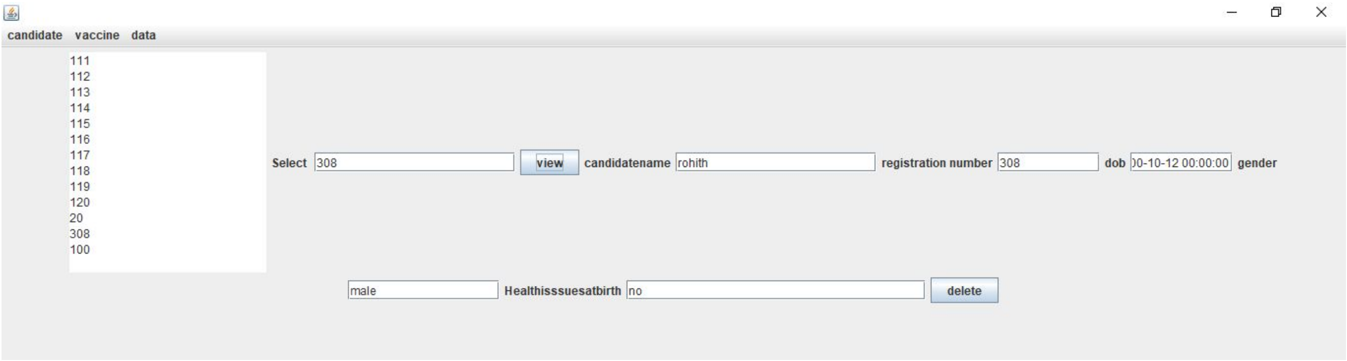
NOW PERFORMING DELETE OPERATION CANDIDATES TABLE



111
112
113
114
115
116
117
118
119
120
20
308
100

Select candidatename registration number dob gender Healthisssuesatbirth

Title:SUGGESTION BASED IMMUNE VACCINATION MANAGEMENT SYSTEM.



111
112
113
114
115
116
117
118
119
120
20
308
100

Select: 308

candidatename: rohith

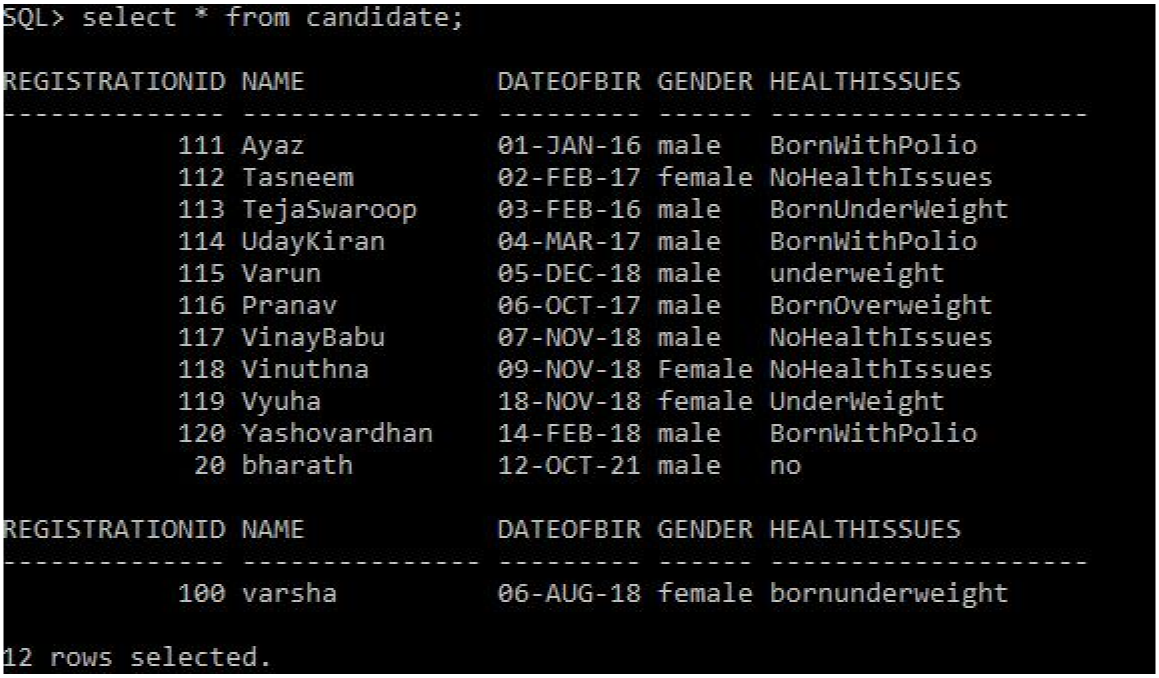
registration number: 308

dob: 00-10-12 00:00:00

gender: male

Healthisssuesatbirth: no

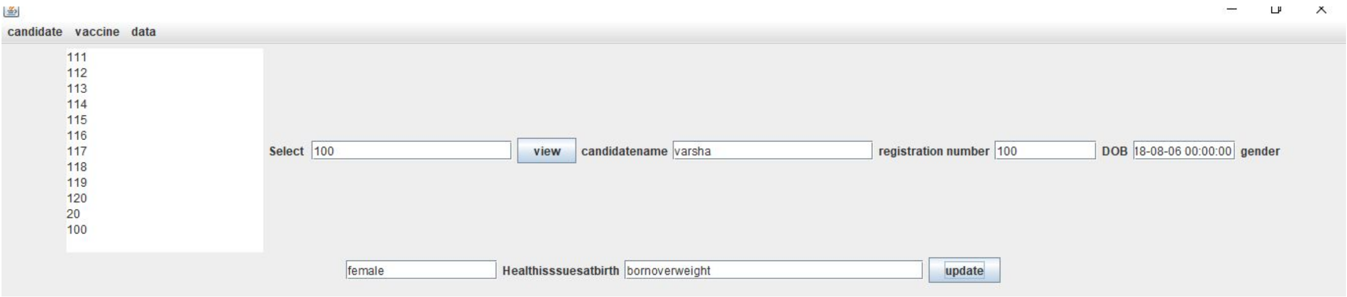
DELETING 308'S RECORD AFTER CHECKING HIS DETAILS



```
SQL> select * from candidate;

REGISTRATIONID NAME            DATEOFBIR GENDER HEALTHISSUES
-------------- --------------- --------- ------ --------------------
           111 Ayaz            01-JAN-16 male   BornWithPolio
           112 Tasneem         02-FEB-17 female NoHealthIssues
           113 TejaSwaroop     03-FEB-16 male   BornUnderWeight
           114 UdayKiran       04-MAR-17 male   BornWithPolio
           115 Varun           05-DEC-18 male   underweight
           116 Pranav          06-OCT-17 male   BornOverweight
           117 VinayBabu       07-NOV-18 male   NoHealthIssues
           118 Vinuthna        09-NOV-18 Female NoHealthIssues
           119 Vyuha           18-NOV-18 female UnderWeight
           120 Yashovardhan    14-FEB-18 male   BornWithPolio
            20 bharath         12-OCT-21 male   no

REGISTRATIONID NAME            DATEOFBIR GENDER HEALTHISSUES
-------------- --------------- --------- ------ --------------------
           100 varsha          06-AUG-18 female bornunderweight

12 rows selected.
```

RECORD WAS DELETE ON DATABASE

NOW LET US UPDATE DATA



111
112
113
114
115
116
117
118
119
120
20
100

Select 100

candidatename varsha

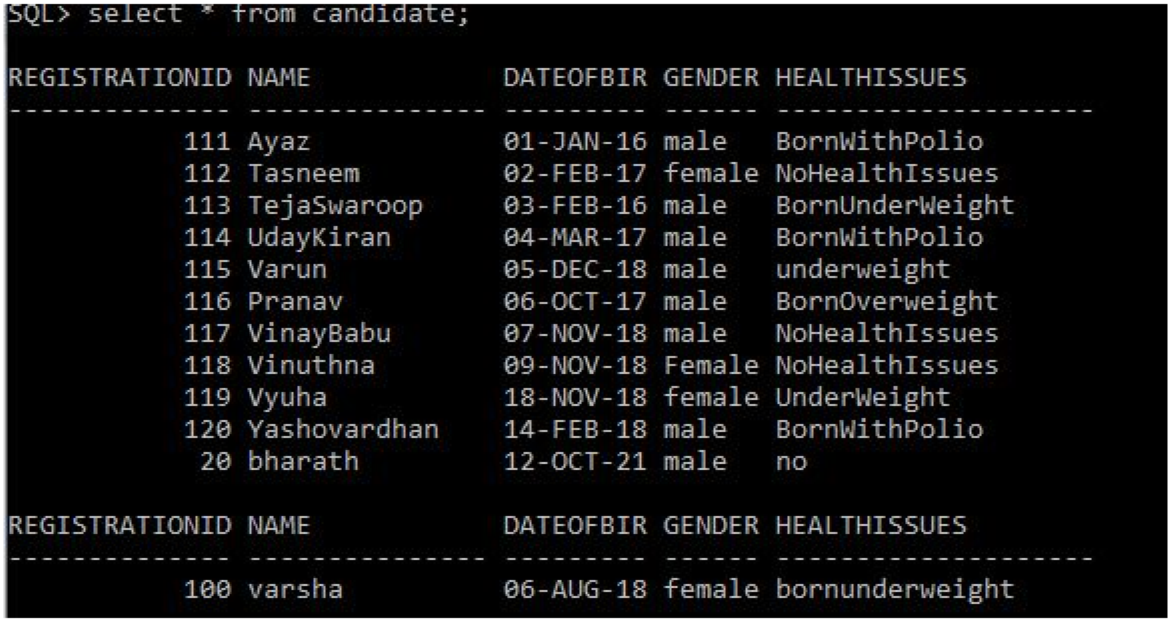
registration number 100

DOB 18-08-06 00:00:00

gender female

Healthisssuesatbirth bornoverweight

100’s healthissue is updated



```
SQL> select * from candidate;

REGISTRATIONID NAME            DATEOFBIR GENDER HEALTHISSUES
-------------- --------------- --------- ------ --------------------
           111 Ayaz            01-JAN-16 male   BornWithPolio
           112 Tasneem         02-FEB-17 female NoHealthIssues
           113 TejaSwaroop     03-FEB-16 male   BornUnderWeight
           114 UdayKiran       04-MAR-17 male   BornWithPolio
           115 Varun           05-DEC-18 male   underweight
           116 Pranav          06-OCT-17 male   BornOverweight
           117 VinayBabu       07-NOV-18 male   NoHealthIssues
           118 Vinuthna        09-NOV-18 Female NoHealthIssues
           119 Vyuha           18-NOV-18 female UnderWeight
           120 Yashovardhan    14-FEB-18 male   BornWithPolio
            20 bharath         12-OCT-21 male   no

REGISTRATIONID NAME            DATEOFBIR GENDER HEALTHISSUES
-------------- --------------- --------- ------ --------------------
           100 varsha          06-AUG-18 female bornunderweight
```

CONCLUSION:

with the help of the project,proper vaccine can be taken at proper time for their problem

GITHUB LINK:

https://github.com/varunbelide777/DBMSMINIPROJECT