**Lab 27**

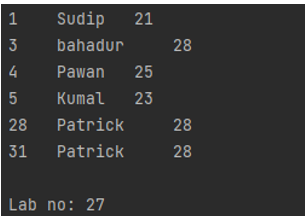
WAP in Java to read and display all the data stored in a table of some database using JDBC.(use Statement interface).

JDBC (Java Database Connectivity) is the Java API that manages connecting to a database, issuing queries and commands, and handling result sets obtained from the database. A ResultSet object maintains a cursor that points to the current row in the result set. The term "result set" refers to the row and column data contained in a ResultSet object.

**Code:**

import java.sql.\*;  
public class Main {  
 public static void main(String[] args) throws ClassNotFoundException, SQLException {  
 Class.*forName*("com.mysql.cj.jdbc.Driver");  
 String url = "jdbc:mysql://localhost:3306/dab";  
 String user = "root";  
 String password = "admin";  
 String query = "SELECT \* FROM mytable";  
 Connection conn = DriverManager.*getConnection*(url, user, password);  
 Statement stmt = conn.createStatement();  
 ResultSet rs = stmt.executeQuery(query);  
 while (rs.next()) {  
 System.*out*.println(rs.getInt("id") + "\t " + rs.getString("uname") + "\t " +  
 rs.getInt("age"));  
 }  
 rs.close();  
 stmt.close();  
 conn.close();  
 System.*out*.println("\nLab no: 27 \nName: Sudip Shrestha \nRoll No/Section: 20/A");  
 }  
}

**Output:**



**Lab 28**

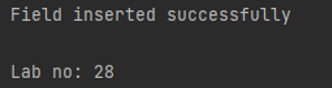
Write a program using JDBC to insert a row into a table of some database. (use Statement interface).

JDBC (Java Database Connectivity) is the Java API that manages connecting to a database, issuing queries and commands, and handling result sets obtained from the database. The statement interface is used to create SQL basic statements in Java it provides methods to execute queries with the database.

**Code:**

import java.sql.\*;  
public class Main {  
 public static void main(String[] args) throws ClassNotFoundException, SQLException {  
 Class.*forName*("com.mysql.cj.jdbc.Driver");  
 String url = "jdbc:mysql://localhost:3306/dab";  
 String user = "root";  
 String password = "admin";  
 String query = "Insert into mytable values(90,'Sudip',23)";  
 Connection conn = DriverManager.*getConnection*(url, user, password);  
 Statement stmt = conn.createStatement();  
 int n = stmt.executeUpdate(query);  
 if(n>0){  
 System.*out*.println("Field inserted successfully");  
 }  
 stmt.close();  
 conn.close();  
 System.*out*.println("\nLab no: 28 \nName: Sudip Shrestha \nRoll No/Section: 20/A");  
 }  
}

**Output:**



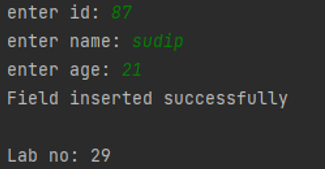
**Lab 29**

Write a java program that allows a user to insert values to a table of particular database (Suppose database is in MySql server).

**Code:**

import java.sql.\*;  
import java.util.Scanner;  
public class Main {  
 public static void main(String[] args) throws ClassNotFoundException, SQLException {  
 Class.*forName*("com.mysql.cj.jdbc.Driver");  
 Scanner sc1=new Scanner(System.*in*);  
 String url = "jdbc:mysql://localhost:3306/dab";  
 String user = "root";  
 String password = "admin";  
 System.*out*.print("enter id: ");  
 int id=sc1.nextInt();  
 System.*out*.print("enter name: ");  
 String name=sc1.next();  
 System.*out*.print("enter age: ");  
 int age=sc1.nextInt();  
 String query = "Insert into mytable values("+id+",'"+name+"',"+age+")";  
 Connection conn = DriverManager.*getConnection*(url, user, password);  
 Statement stmt = conn.createStatement();  
 int n = stmt.executeUpdate(query);  
 if(n>0){  
 System.*out*.println("Field inserted successfully");  
 }  
 stmt.close();  
 conn.close();  
 System.*out*.println("\nLab no: 29 \nName: Sudip Shrestha \nRoll No/Section: 20/A");  
 }  
}

**Output:**



**Lab 30**

WAP using JDBC to display the records from a table of given database (Suppose database is in MySql server). Assume the following table :

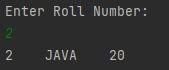
result(roll\_no , course\_name ,marks\_obtained)

The program should read the roll number value from console and display the corresponding record.

**Code:**

import java.sql.\*;  
public class Main {  
 public static void main(String[] args) throws ClassNotFoundException, SQLException {  
 Class.*forName*("com.mysql.cj.jdbc.Driver");  
 String url = "jdbc:mysql://localhost:3306/dab";  
 String user = "root";  
 String password = "admin";  
 String query = "SELECT \* FROM mytable";  
 Connection conn = DriverManager.*getConnection*(url, user, password);  
 Statement stmt = conn.createStatement();  
 ResultSet rs = stmt.executeQuery(query);  
 while (rs.next()) {  
 System.*out*.println(rs.getInt("id") + "\t " + rs.getString("uname") + "\t " +  
 rs.getInt("age"));  
 }  
 rs.close();  
 stmt.close();  
 conn.close();  
 System.*out*.println("\nLab no: 27 \nName: Sudip Shrestha \nRoll No/Section: 20/A");  
 }  
}

**Output:**



**Lab 31**

WAP using JDBC to

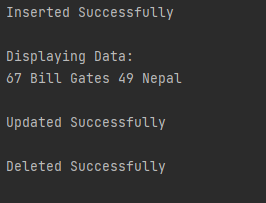
* insert a record into table using PreparedStatement.
* retrieve and display the records of table using PreparedStatement.
* delete a record of a table using PreparedStatement.
* update the record of table using PreparedStatement.

A PreparedStatement is a pre-compiled SQL statement. It is a subinterface of Statement. Prepared Statement objects have some useful additional features than Statement objects. Instead of hard coding queries, PreparedStatement object provides a feature to execute a parameterized query.

**Code:**

import java.sql.\*;  
public class Main {  
 public static void main(String[] args) throws ClassNotFoundException, SQLException {  
 Class.*forName*("com.mysql.cj.jdbc.Driver");  
 String url = "jdbc:mysql://localhost:3306/dab";  
 String user = "root";  
 String password = "admin";  
 Connection conn = DriverManager.*getConnection*(url, user, password);  
 String query="Insert into employee values (?,?,?,?)";  
 PreparedStatement ps= conn.prepareStatement(query);  
 ps.setInt(1,79);  
 ps.setString(2,"Bill Gates");  
 ps.setInt(3,34);  
 ps.setString(4,"US");  
 int n=ps.executeUpdate();  
 if(n>0){  
 System.*out*.println("Inserted Successfully \n");  
 }  
 String query2="Select\* from employee where empid=?";  
 PreparedStatement ps2= conn.prepareStatement(query2);  
 ps2.setInt(1,67);  
 ResultSet rs=ps2.executeQuery();  
 System.*out*.println("Displaying Data:");  
 while(rs.next()){  
 System.*out*.println(rs.getInt("empid")+" "+rs.getString("empnme")+" "+rs.getInt("empage")+" "+rs.getString("empaddress"));  
 }  
 System.*out*.println();  
 String query3="Update Employee set empnme=?,empage=?,empaddress=? where empid=?";  
 PreparedStatement ps3= conn.prepareStatement(query3);  
 ps3.setInt(4,67);  
 ps3.setString(1,"Bill Gates");  
 ps3.setInt(2,49);  
 ps3.setString(3,"Nepal");  
 int n2=ps3.executeUpdate();  
 if(n2>0){  
 System.*out*.println("Updated Successfully \n");  
 }  
 String query4="Delete from employee where empid=?";  
 PreparedStatement ps4= conn.prepareStatement(query4);  
 ps4.setInt(1,67);  
 int n3=ps4.executeUpdate();  
 if(n3>0){  
 System.*out*.println("Deleted Successfully \n");  
 }  
 ps.close();  
 conn.close();  
 }  
}

**Output:**



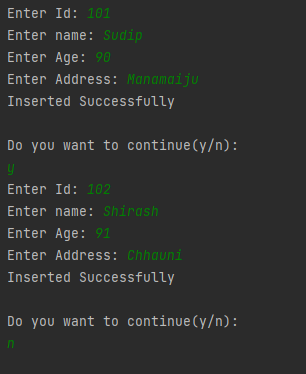
**Lab 32**

Write a java program using PreparedStatement that allows a user to insert values to a table of particular database (Suppose database is in MySql server ). The program should take the values to insert from console as long as user want to add new record.

**Code:**

import java.sql.\*;  
import java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) throws ClassNotFoundException, SQLException {  
 Class.*forName*("com.mysql.cj.jdbc.Driver");  
 Scanner sc1=new Scanner(System.*in*);  
 String url = "jdbc:mysql://localhost:3306/dab";  
 String user = "root";  
 String password = "admin";  
 char switch1='y';  
 int id=0;  
 int age=0;  
 String name="";  
 String address="";  
 PreparedStatement ps=null;  
 Connection conn = DriverManager.*getConnection*(url, user, password);  
 do{  
 System.*out*.print("Enter Id: ");  
 id=sc1.nextInt();  
 System.*out*.print("Enter name: ");  
 name=sc1.next();  
 System.*out*.print("Enter Age: ");  
 age=sc1.nextInt();  
 System.*out*.print("Enter Address: ");  
 address=sc1.next();  
 String query2="Insert into employee values (?,?,?,?)";  
 ps= conn.prepareStatement(query2);  
 ps.setInt(1,id);  
 ps.setString(2,name);  
 ps.setInt(3,age);  
 ps.setString(4,address);  
 int n=ps.executeUpdate();  
 if(n>0){  
 System.*out*.println("Inserted Successfully \n");  
 }  
 System.*out*.println("Do you want to continue(y/n): ");  
 switch1=sc1.next().charAt(0);  
 }while(switch1=='y');  
 ps.close();  
 conn.close();  
 }  
}

**Output:**



**Lab 33**

WAP using PreparedStatement to display the records from a table of given database (Suppose database is in MySql server ). Assume the following table : salary(emp\_id , emp\_name ,emp\_salary) The program should read the employee id valuefrom console and display the corresponding record.

**Code:**

import java.sql.\*;  
import java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) throws ClassNotFoundException, SQLException {  
 Class.*forName*("com.mysql.cj.jdbc.Driver");  
 Scanner sc1=new Scanner(System.*in*);  
 String url = "jdbc:mysql://localhost:3306/dab";  
 String user = "root";  
 String password = "admin";  
 PreparedStatement ps=null;  
 Connection conn = DriverManager.*getConnection*(url, user, password);  
 System.*out*.print("Enter Id: ");  
 int id=sc1.nextInt();  
 String query2="Select\* from salary where emp\_id=? ";  
 ps= conn.prepareStatement(query2);  
 ps.setInt(1,id);  
 ResultSet rs=ps.executeQuery();  
 while(rs.next()){  
 System.*out*.println(rs.getInt("emp\_id")+" "+rs.getString("emp\_name")+" "+rs.getInt("emp\_salary"));  
 }  
 ps.close();  
 conn.close();  
 }  
}

**Output:**



**Lab 34**

WAP using JDBC to demonstrate the concept of:

a) Scrollable ResultSet

b) Updatable ResultSet

c) Multiple ResultSet

A scrollable ResultSet is one which allows us to retrieve the data in forward direction as well as backward direction but no updations are allowed. Whenever we create a ResultSet object which never allows us to update the database through ResultSet object and it allows retrieving the data only in forward direction. Such type of ResultSet is known as non-updatable and non-scrollable ResultSet.

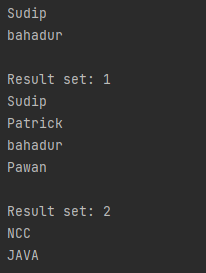
In order to make the ResultSet object as updatable and scrollable we must use the following constants which are present in ResultSet interface.

The above two constants must be specified while we are creating Statement object

**Code:**

import java.sql.\*;  
import java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) throws ClassNotFoundException, SQLException {  
 Class.*forName*("com.mysql.cj.jdbc.Driver");  
 Scanner sc1=new Scanner(System.*in*);  
 String url = "jdbc:mysql://localhost:3306/dab";  
 String user = "root";  
 String password = "admin";  
 PreparedStatement ps=null;  
 Connection conn = DriverManager.*getConnection*(url, user, password);  
 Statement st=conn.createStatement(ResultSet.*TYPE\_SCROLL\_SENSITIVE*,ResultSet.*CONCUR\_UPDATABLE*);  
 ResultSet rs=st.executeQuery("Select\* from mytable");  
 rs.absolute(1);  
 System.*out*.println(rs.getString("uname"));  
 rs.moveToInsertRow();  
 rs.updateInt("id",2);  
 rs.updateString("uname","Patrick");  
 rs.updateInt("age",28);  
 rs.insertRow();  
 rs.absolute(2);  
 System.*out*.println(rs.getString("uname"));  
 rs.updateString("uname","bahadur");  
 rs.updateRow();  
 Statement stmt = conn.createStatement();  
 String query1 = "SELECT \* FROM mytable";  
 String query2 = "SELECT \* FROM mytable2";  
 boolean hasMoreResults = stmt.execute(query1);  
 int resultSetCount = 0;  
 while (hasMoreResults) {  
 resultSetCount++;  
 System.*out*.println("\nResult set: " + resultSetCount);  
 ResultSet rs2 = stmt.getResultSet();  
 while (rs2.next()) {  
 System.*out*.println(rs2.getString(2));  
 }  
  
 hasMoreResults = stmt.getMoreResults();  
 }  
 boolean hasMoreResults1 = stmt.execute(query2);  
 while (hasMoreResults1) {  
 resultSetCount++;  
 System.*out*.println("\nResult set: " + resultSetCount);  
 ResultSet rs2 = stmt.getResultSet();  
 while (rs2.next()) {  
 System.*out*.println(rs2.getString(2));  
 }  
  
 hasMoreResults1 = stmt.getMoreResults();  
 }  
 stmt.close();  
 rs.close();  
 st.close();  
 conn.close();  
 }  
}

**Output:**



**Lab 35**

WAP using JDBC to illustrate the concept of:

a) JdbcRowSet

b) CachedRowSet

JDBC (Java Database Connectivity) is a Java API that provides a standard way of accessing databases from Java programs. A RowSet is an extension of the ResultSet interface in JDBC that allows disconnected, scrollable, and updatable access to the result set.

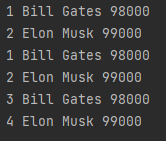
A CachedRowSet is a subclass of RowSet that provides an implementation of the RowSet interface that caches all of its rows in memory. This allows the CachedRowSet to be disconnected from the database after it has been populated with data, providing better performance and scalability for applications that require frequent access to the same data.

1. JDBC Rowset

**Code:**

import javax.sql.rowset.JdbcRowSet;  
import javax.sql.rowset.RowSetProvider;  
  
public class Main {  
 public static void main(String[] args) throws Exception {  
 Class.*forName*("com.mysql.cj.jdbc.Driver");  
 JdbcRowSet rowSet = RowSetProvider.*newFactory*().createJdbcRowSet();  
 rowSet.setUrl("jdbc:mysql://localhost:3306/dab");  
 rowSet.setUsername("root");  
 rowSet.setPassword("admin");  
 rowSet.setCommand("select \* from salary");  
 rowSet.execute();  
 while (rowSet.next()) {  
 System.*out*.println(rowSet.getInt(1)+" "+rowSet.getString(2)+" "+rowSet.getInt(3));  
 }  
 }  
}

**Output:**

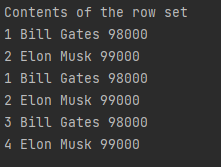


1. Cached Rowset

**Code:**

import javax.sql.rowset.\*;  
public class Main {  
 public static void main(String args[]) throws Exception {  
 RowSetFactory factory = RowSetProvider.*newFactory*();  
 CachedRowSet rowSet = factory.createCachedRowSet();  
 rowSet.setUrl("jdbc:mysql://localhost:3306/dab");  
 rowSet.setUsername("root");  
 rowSet.setPassword("admin");  
 rowSet.setCommand("select \* from salary");  
 rowSet.execute();  
 System.*out*.println("Contents of the row set");  
 while(rowSet.next()) {  
 System.*out*.println(rowSet.getInt(1)+" "+rowSet.getString(2)+" "+rowSet.getInt(3));  
 }  
 }  
}

**Output:**



**Lab 36**

WAP to demonstrate transaction management in JDBC.

**Code:**

import java.sql.\*;

import java.util.Scanner;

public class Main {

public static void main(String[] args) throws ClassNotFoundException, SQLException {

Class.forName("com.mysql.cj.jdbc.Driver");

Scanner sc1=new Scanner(System.in);

String url = "jdbc:mysql://localhost:3306/dab";

String user = "root";

String password = "admin";

PreparedStatement ps=null;

Connection conn = DriverManager.getConnection(url, user, password);

conn.setAutoCommit(false);

Statement st =null;

try {

st = conn.createStatement();

String query = "Insert into salary(11,'Bill',2000";

st.executeUpdate(query);

String query2 = "Insert into salary(11,'Nill','feel'";

st.executeUpdate(query2);

String query3 = "Insert into salary(14,'Hill',2000";

st.executeUpdate(query3);

conn.commit();

}catch(Exception ex){

conn.rollback();

System.out.println("Transaction rollback completed");

}

st.close();

conn.close();

}

}

Output:



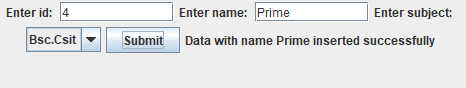
**Lab 37**

Develop a CRUD application using swing and JDBC. Your UI must contain text field, radio button and combo box (use other components as required). [source code can be printed for this task]

Create

import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.sql.\*;  
public class Main extends JFrame implements ActionListener {  
 JLabel l1 = new JLabel("Enter id: ");  
 JLabel l2 = new JLabel("Enter name: ");  
 JLabel l3 = new JLabel("Enter subject: ");  
 JTextField t1 = new JTextField(10);  
 JTextField t2 = new JTextField(10);  
 String[] st = new String[]{"Bsc.Csit", "BCA", "BIM"};  
 JComboBox jc = new JComboBox(st);  
 JButton b1 = new JButton("Submit");  
 JLabel l4=new JLabel("");  
 public Main() {  
 setTitle("Lab 37");  
 setLayout(new FlowLayout(1));  
 setSize(500, 500);  
 add(l1);  
 add(t1);  
 add(l2);  
 add(t2);  
 add(l3);  
 add(jc);  
 add(b1);  
 add(l4);  
 b1.addActionListener(this);  
 setVisible(true);  
 setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  
 }  
  
 public static void main(String[] args) {  
 new Main();  
 }  
  
 @Override  
 public void actionPerformed(ActionEvent e){  
 Connection conn = null;  
 Statement st = null;  
 ResultSet rs = null;  
 try{  
 Class.*forName*("com.mysql.cj.jdbc.Driver");  
 String url = "jdbc:mysql://localhost:3306/dab";  
 String uname = "root";  
 String pw = "admin";  
 conn = DriverManager.*getConnection*(url, uname, pw);  
 int id = Integer.*parseInt*(t1.getText());  
 String name = t2.getText();  
 String subject=(String)jc.getSelectedItem();  
 String query = "Insert into User1 VALUES(" + id + ",'" + name + "','" + subject + "')";  
 st = conn.createStatement();  
 int a = st.executeUpdate(query);  
 if (a > 0) {  
 l4.setText("Data with name "+name+" inserted successfully");  
 } else {  
 System.*out*.println("Could not Insert, Sorry :(");  
 }  
 st.close();  
 } catch (SQLException ex) {  
 throw new RuntimeException(ex);  
 } catch (ClassNotFoundException ex) {  
 throw new RuntimeException(ex);  
 }  
 }  
}

**Output:**



Read

import javax.swing.\*;  
import java.awt.\*;  
import java.sql.\*;  
public class Main extends JFrame{  
 JLabel l1=new JLabel("");  
 public Main() {  
 setTitle("Lab 37");  
 setLayout(new FlowLayout(1));  
 setSize(500, 500);  
 add(l1);  
 setVisible(true);  
 setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);  
 }  
 public static void main(String[] args) {  
 Main m1 = new Main();  
 m1.my1();  
 }  
 public void my1(){  
 Connection conn = null;  
 Statement st = null;  
 ResultSet rs = null;  
 try {  
 Class.forName("com.mysql.cj.jdbc.Driver");  
 String url = "jdbc:mysql://localhost:3306/dab";  
 String uname = "root";  
 String pw = "admin";  
 conn = DriverManager.getConnection(url, uname, pw);  
 String query = "Select \* from User1";  
 st = conn.createStatement();  
 rs = st.executeQuery(query);  
 String st1="";  
 while(rs.next()){  
 st1+=(rs.getInt("id")+" "+rs.getString("Uname")+" "+rs.getString("Usubject")+" ");  
 }  
 l1.setText(st1);  
 st.close();  
 conn.close();  
 } catch (Exception ex) {  
 throw new RuntimeException(ex);  
 }  
 }  
}

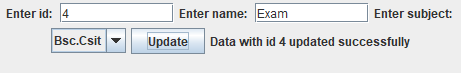
Output:



Update:

import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.sql.\*;  
public class Main extends JFrame implements ActionListener {  
 JLabel l1 = new JLabel("Enter id: ");  
 JLabel l2 = new JLabel("Enter name: ");  
 JLabel l3 = new JLabel("Enter subject: ");  
 JTextField t1 = new JTextField(10);  
 JTextField t2 = new JTextField(10);  
 String[] st = new String[]{"Bsc.Csit", "BCA", "BIM"};  
 JComboBox jc = new JComboBox(st);  
 JButton b1 = new JButton("Update");  
 JLabel l4=new JLabel("");  
 public Main() {  
 setTitle("Lab 37");  
 setLayout(new FlowLayout(1));  
 setSize(500, 500);  
 add(l1);  
 add(t1);  
 add(l2);  
 add(t2);  
 add(l3);  
 add(jc);  
 add(b1);  
 add(l4);  
 b1.addActionListener(this);  
 setVisible(true);  
 setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  
 }  
 public static void main(String[] args) {  
 new Main();  
 }  
 @Override  
 public void actionPerformed(ActionEvent e){  
 Connection conn = null;  
 Statement st = null;  
 ResultSet rs = null;  
 try{  
 Class.*forName*("com.mysql.cj.jdbc.Driver");  
 String url = "jdbc:mysql://localhost:3306/dab";  
 String uname = "root";  
 String pw = "admin";  
 conn = DriverManager.*getConnection*(url, uname, pw);  
 int id = Integer.*parseInt*(t1.getText());  
 String name = t2.getText();  
 String subject=(String)jc.getSelectedItem();  
 String query = "Update User1 set Uname='"+name+"',Usubject='"+subject+"' where id="+id;  
 st = conn.createStatement();  
 int a = st.executeUpdate(query);  
 if (a > 0) {  
 l4.setText("Data with id "+id+" updated successfully");  
 } else {  
 System.*out*.println("Could not Update, Sorry :(");  
 }  
 st.close();  
 } catch (SQLException ex) {  
 throw new RuntimeException(ex);  
 } catch (ClassNotFoundException ex) {  
 throw new RuntimeException(ex);  
 }  
 }  
}

**Output:**



Delete

import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.sql.\*;  
public class Main extends JFrame implements ActionListener {  
 JLabel l1 = new JLabel("Enter id you want to delete: ");  
 JTextField t1 = new JTextField(10);  
 JButton b1 = new JButton("Delete");  
 JLabel l4=new JLabel("");  
 public Main() {  
 setTitle("Lab 37");  
 setLayout(new FlowLayout(1));  
 setSize(500, 500);  
 add(l1);  
 add(t1);  
 add(b1);  
 add(l4);  
 b1.addActionListener(this);  
 setVisible(true);  
 setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  
 }  
 public static void main(String[] args) {  
 new Main();  
 }  
 @Override  
 public void actionPerformed(ActionEvent e){  
 Connection conn = null;  
 Statement st = null;  
 ResultSet rs = null;  
 try{  
 Class.*forName*("com.mysql.cj.jdbc.Driver");  
 String url = "jdbc:mysql://localhost:3306/dab";  
 String uname = "root";  
 String pw = "admin";  
 conn = DriverManager.*getConnection*(url, uname, pw);  
 int id = Integer.*parseInt*(t1.getText());  
 String query = "Delete from User1 where id="+id;  
 st = conn.createStatement();  
 int a = st.executeUpdate(query);  
 if (a > 0) {  
 l4.setText("Data with id "+id+" Deleted successfully");  
 } else {  
 System.*out*.println("Could not Delete, Sorry :(");  
 }  
 st.close();  
 } catch (SQLException ex) {  
 throw new RuntimeException(ex);  
 } catch (ClassNotFoundException ex) {  
 throw new RuntimeException(ex);  
 }  
 }  
}

**Output:**

