TRINITY INTERNATIONAL COLLEGE

(Tribhuvan University Affiliated)

Lab Report:4 JDBC

Submitted by:

Name :Anusha Panta
Program : **B. Sc. (CSIT)**
Roll No :10
Semester: 7th
Date :21/06/2020

Submitted to:

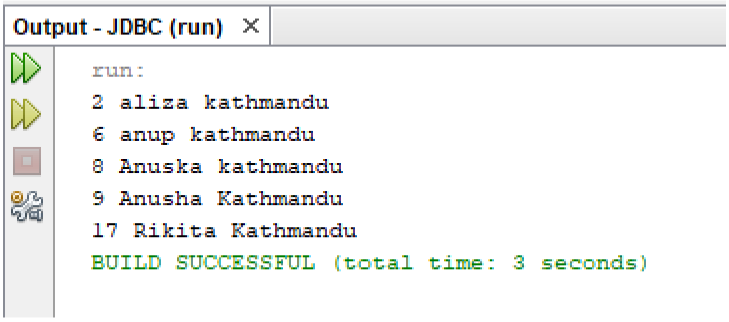
Aman Maharjan

KATHMANDU, NEPAL
2020

Unit 4: Database Connectivity

1. **Write a Java program using JDBC to extract name of those students who live in Kathmandu district, assuming that the student table has four attributes (ID, name, district, and age). [2072]**

```
package Q1_SelectName;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

/**
 *
 * @author user
 */
public class SelectStudentsName {
    public static void main(String[] args) throws SQLException {
        String url = "jdbc:mariadb://localhost:3306/students";
        String username = "root";
        String password = "";
        Connection connection = DriverManager.getConnection(url,
         username, password);
        Statement statement = (Statement)
             connection.createStatement();
        String sql = "select * from studentdata where
            district='kathmandu' ";
        ResultSet rs = statement.executeQuery(sql);
        while (rs.next()) {
            System.out.printf("%d %s %s \n",
                    rs.getInt("id"),
                    rs.getString("name"),
                    rs.getString("district"),
                    rs.getInt("age"));
        }

        statement.close();
        connection.close();
    }
}
```

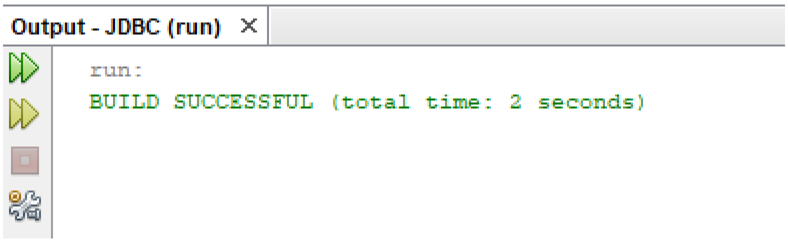


```
run:
2 aliza kathmandu
6 anup kathmandu
8 Anuska kathmandu
9 Anusha Kathmandu
17 Rikita Kathmandu
BUILD SUCCESSFUL (total time: 3 seconds)
```

2. Write a program to illustrate the process of executing SQL statements in JDBC? [2073, 2074]

SQLINSERT

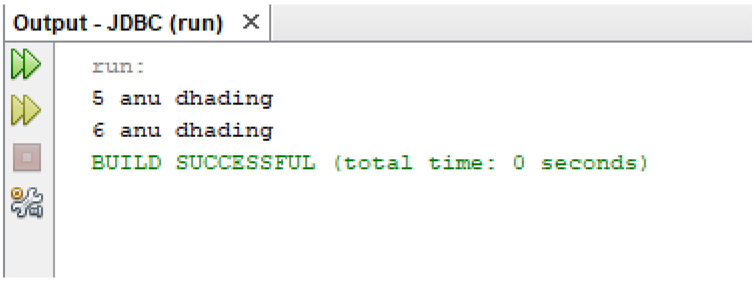
```
package Q2_SQLLstatement;

import java.sql.Statement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;

/**
 *
 * @author user
 */
public class SqlInsertDemo {

    public static void main(String[] args) throws SQLException {
        String url = "jdbc:mariadb://localhost:3306/java";
        String username = "root";
        String password = "";
        Connection connection = DriverManager.getConnection(url,
username, password);
        Statement statement = (Statement)
connection.createStatement();
        String sql = "insert into contacts (name,address) " +
"values('anu','dhading')";
        ResultSet rs = statement.executeQuery(sql);

        statement.close();
        connection.close();

    }

}
```



```
run:
BUILD SUCCESSFUL (total time: 2 seconds)
```

SQLSELECT

```
package Q2_SQLLstatement;

import java.sql.Statement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;

/**
 *
 * @author user
 */
public class SqlSelectDemo {

    public static void main(String[] args) throws SQLException {
        String url = "jdbc:mariadb://localhost:3306/java";
        String username = "root";
        String password = "";
        Connection connection = DriverManager.getConnection(url,
username, password);
        Statement statement = (Statement)
connection.createStatement();
        String sql = "select * from contacts where name='anu' ";
        ResultSet rs = statement.executeQuery(sql);
        while (rs.next()) {
            System.out.printf("%d %s %s \n",
                    rs.getInt("id"),
                    rs.getString("name"),
                    rs.getString("address"));
        }

        statement.close();
        connection.close();
    }

}
```

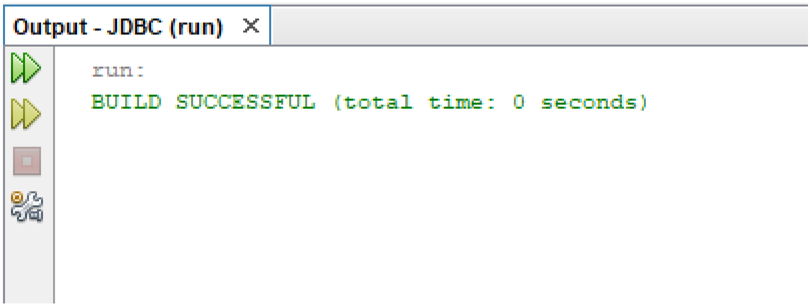


```
run:
5 anu dhading
6 anu dhading
BUILD SUCCESSFUL (total time: 0 seconds)
```

SQLUPDATE

```
package Q2_SQLLstatement;

import java.sql.Statement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;

/**
 *
 * @author user
 */
```

```
public class SqlUpdateDemo {

    public static void main(String[] args) throws SQLException {
        String url = "jdbc:mariadb://localhost:3306/java";
        String username = "root";
        String password = "";
        Connection connection = DriverManager.getConnection(url,
username, password);
        Statement statement = (Statement)
connection.createStatement();
        String sql = "update contacts set address='dhading' where id
= 1";
        ResultSet rs = statement.executeQuery(sql);

        statement.close();
        connection.close();

    }

}
```

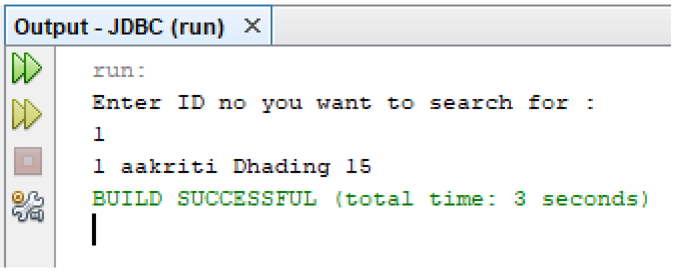


```
run:
BUILD SUCCESSFUL (total time: 0 seconds)
```

3. Implement CRUD (Create/Insert, Read/Select, Update, Delete) operations for student table. Ask for user input where applicable.

SelectStudentData.java

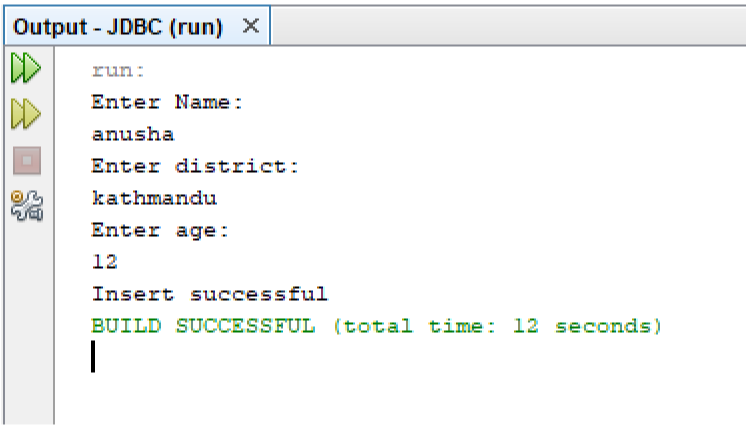
```
package Q3_CRUD_SQLStatement;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;

/**
 *
 * @author user
 */
public class SelectStudentData {

    public static void main(String[] args) throws SQLException {
        String url = "jdbc:mariadb://localhost:3306/students";
        String username = "root";
        String password = "";
        Connection connection = DriverManager.getConnection(url,
username, password);
        Statement statement = (Statement)
connection.createStatement();
        Scanner s = new Scanner(System.in);
        System.out.println("Enter ID no you want to search for :");
        int Id = s.nextInt();
        String sql = String.format(
                "select * from  studentdata where id='%d '", Id
        );
        ResultSet rows = statement.executeQuery(sql);
        while (rows.next()) {
            System.out.printf("%d %s %s %d \n",
                    rows.getInt("id"),
                    rows.getString("name"),
                    rows.getString("district"),
                    rows.getInt("age"));
        }
        statement.close();
        connection.close();
    }

}
```



```
run:
Enter ID no you want to search for :
1
1 aakriti Dhading 15
BUILD SUCCESSFUL (total time: 3 seconds)
```

InsertStudentsData

```
package Q3_CRUD_SQLStatement;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;

/**
 *
 * @author user
 */
public class InsertStudentsData {
    public static void main(String[] args) throws SQLException {
        String url = "jdbc:mariadb://localhost:3306/students";
        String username = "root";
        String password = "";
        Connection connection = DriverManager.getConnection(url,
username, password);
        Statement statement = (Statement)
connection.createStatement();
        Scanner s=new Scanner(System.in);
        System.out.println("Enter Name:");
        String name=s.nextLine();
        System.out.println("Enter district:");
        String district=s.nextLine();
        System.out.println("Enter age:");
        int age=s.nextInt();
        String sql = String.format(
            "insert into studentdata (name, district,age) values
('%s', '%s','%d')",
            name,
            district,age
        );
        int rowsInserted = statement.executeUpdate(sql);
        if(rowsInserted > 0)
            System.out.println("Insert successful");
        else
            System.out.println("Insert failed");


        statement.close();
        connection.close();
    }
}
```

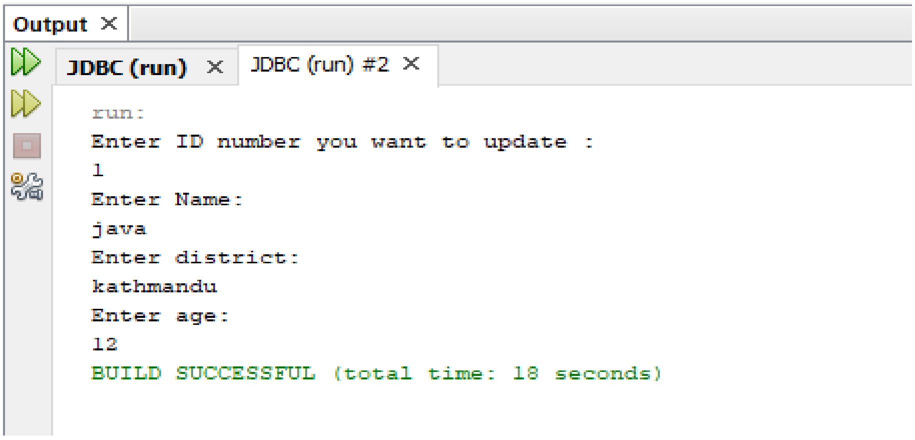


```
run:
Enter Name:
anusha
Enter district:
kathmandu
Enter age:
12
Insert successful
BUILD SUCCESSFUL (total time: 12 seconds)
```

UpdateStudentData

```
package Q3_CRUD_SQLStatement;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;
```

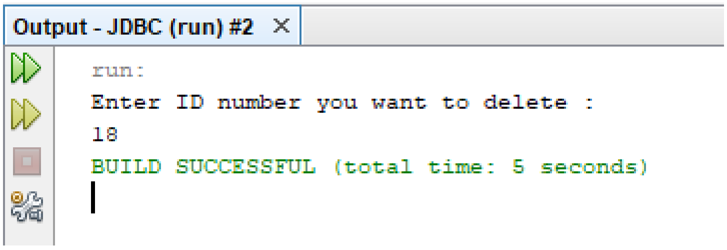
```
/**
 *
 * @author user
 */
public class UpdateStudentData {

    public static void main(String[] args) throws SQLException {
        String url = "jdbc:mariadb://localhost:3306/students";
        String username = "root";
        String password = "";
        Connection connection = DriverManager.getConnection(url,
username, password);
        Statement statement = (Statement)
connection.createStatement();
        Scanner s = new Scanner(System.in);
        System.out.println("Enter ID number you want to update :");
        int id = s.nextInt();

        System.out.println("Enter Name:");
        String name = s.next();
        System.out.println("Enter district:");
        String district = s.next();
        System.out.println("Enter age:");
        int age = s.nextInt();

        String sql = String.format(
                "update studentdata set name='%s',district='%s',"
                + "age='%d' where id = '%d' ", name, district, age,
id
        );
        ResultSet rows = statement.executeQuery(sql);

        statement.close();
        connection.close();
    }

}
```



```
run:
Enter ID number you want to update :
1
Enter Name:
java
Enter district:
kathmandu
Enter age:
12
BUILD SUCCESSFUL (total time: 18 seconds)
```

DeleteStudentsData

```
package Q3_CRUD_SQLStatement;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;

/**
 *
 * @author user
 */
public class DeleteStudentsData {
    public static void main(String[] args) throws SQLException {
        String url = "jdbc:mariadb://localhost:3306/students";
        String username = "root";
        String password = "";
        Connection connection = DriverManager.getConnection(url, 
username, password);
        Statement statement = (Statement) 
connection.createStatement();
        Scanner s = new Scanner(System.in);
        System.out.println("Enter ID number you want to delete :");
        int id = s.nextInt();

        String sql = String.format(
                "delete from studentdata where id = '%d' ", id );
        ResultSet rows = statement.executeQuery(sql);

        statement.close();
        connection.close();
    }
}
```

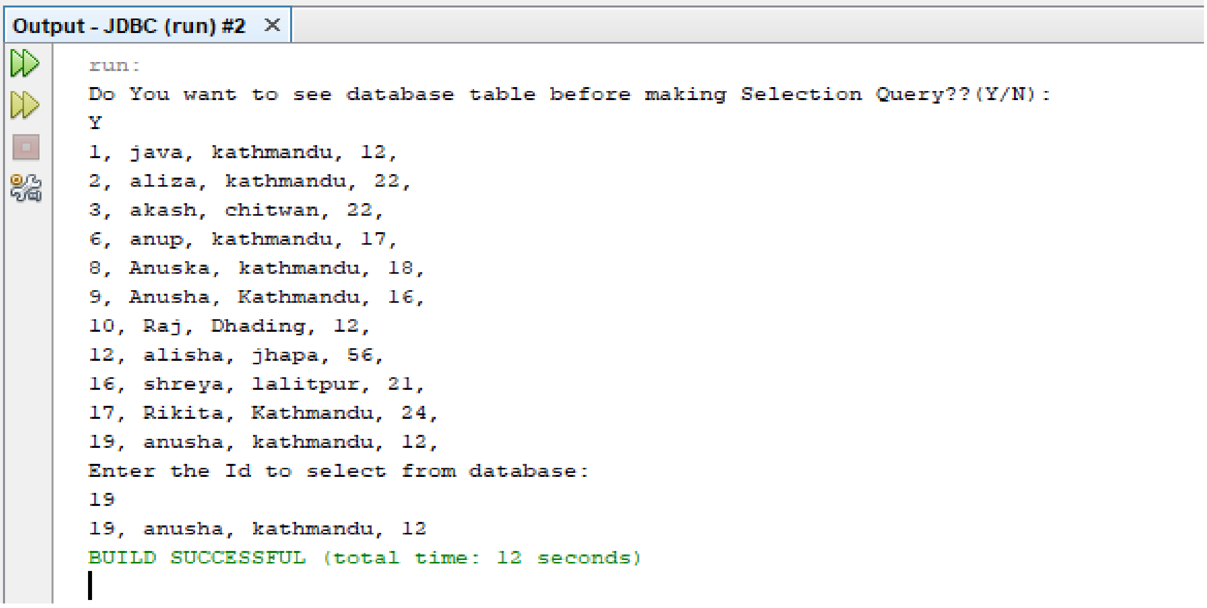


```
run:
Enter ID number you want to delete :
18
BUILD SUCCESSFUL (total time: 5 seconds)
```

4. Implement CRUD operations for student table using prepared statements. Ask for user input where applicable.

SelectPreparedStatement

```
package Q4_CRUD_PreparedStatement;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

/**
 *
 * @author user
 */
public class SelectPreparedStatement {

    public static void main(String[] args) throws SQLException {
//********************************************* Establishing
Connection to the Database
************************************************
        String url = "jdbc:mariadb://localhost:3306/students";
        String username = "root";
        String password = "";
        Connection connection = DriverManager.getConnection(url,
username, password);

// **************** If user want to select but he/she don't know
what values are  inside database table
***********************************
        System.out.println("Do You want to see database table before
making Selection Query??(Y/N):");
        Scanner in = new Scanner(System.in);
        char userChoice = in.next().charAt(0);

        if (userChoice == 'Y') {
            String sql = "select * from studentdata";
            PreparedStatement statement =
connection.prepareStatement(sql);
            ResultSet resultSet = statement.executeQuery();
            while (resultSet.next()) {
                System.out.printf("%d, %s, %s, %d, \n",
                        resultSet.getInt("id"),
                        resultSet.getString("name"),
                        resultSet.getString("district"),
                        resultSet.getInt("age")
                );
            }
            statement.close();
        }

// **************** If user want to select but he/she don't know
what values are  inside database table
***********************************
        if (userChoice == 'N') {
            System.out.println("ok..");
        }
            //************************** Taking user input
*******************************
            System.out.println("Enter the Id to select from
database:");
            int id = in.nextInt();

            //********************** Sql Prepared
Statement************************************
            String sql = "select * from studentdata where id = ?";
            PreparedStatement statement =
connection.prepareStatement(sql);
            statement.setInt(1, id);

            ResultSet resultSet = statement.executeQuery();

            while (resultSet.next()) {
                System.out.printf("%d, %s, %s, %d \n",
                        resultSet.getInt("id"),
                        resultSet.getString("name"),
                        resultSet.getString("district"),
                        resultSet.getInt("age")
                );
            }
            statement.close();


        connection.close();
    }
}
```

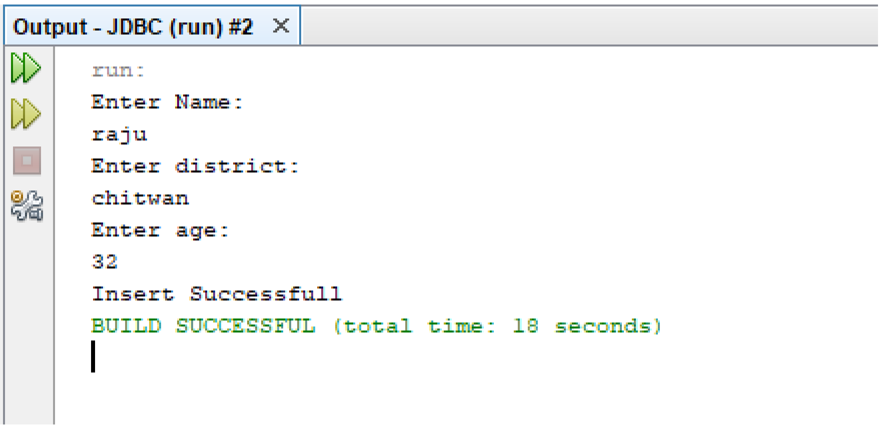


```
run:
Do You want to see database table before making Selection Query??(Y/N):
Y
1, java, kathmandu, 12,
2, aliza, kathmandu, 22,
3, akash, chitwan, 22,
6, anup, kathmandu, 17,
8, Anuska, kathmandu, 18,
9, Anusha, Kathmandu, 16,
10, Raj, Dhading, 12,
12, alisha, jhapa, 56,
16, shreya, lalitpur, 21,
17, Rikita, Kathmandu, 24,
19, anusha, kathmandu, 12,
Enter the Id to select from database:
19
19, anusha, kathmandu, 12
BUILD SUCCESSFUL (total time: 12 seconds)
```

InsertPreparedStatement

```
package Q4_CRUD_PreparedStatement;
```

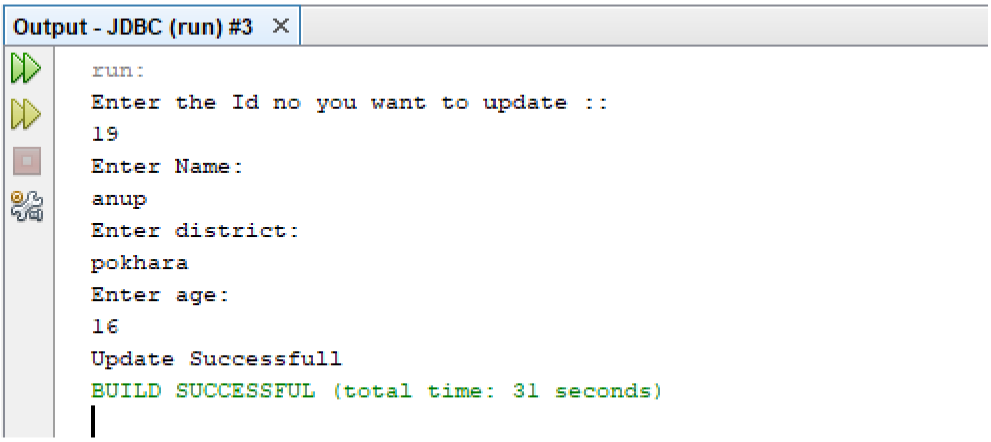
```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

/**
 *
 * @author user
 */
public class InsertPreparedStatement {

    public static void main(String[] args) throws SQLException {
//********************************************* Establishing
Connection to the Database
************************************************
        String url = "jdbc:mariadb://localhost:3306/students";
        String username = "root";
        String password = "";
        Connection connection = DriverManager.getConnection(url,
username, password);

        Scanner s = new Scanner(System.in);
        System.out.println("Enter Name:");
        String name = s.nextLine();
        System.out.println("Enter district:");
        String district = s.nextLine();
        System.out.println("Enter age:");
        int age = s.nextInt();
        String sql = String.format(
                "insert into studentdata (name, district,age) values
('%s', '%s','%d')",
                name,
                district, age
        );

        PreparedStatement statement =
connection.prepareStatement(sql);

        int resultSet = statement.executeUpdate();

        if (resultSet > 0) {
            System.out.println("Insert Successfull");
        } else {
            System.out.println("Insertion Failed");
        }
        statement.close();

        connection.close();
        }

    }
```



```
run:
Enter Name:
raju
Enter district:
chitwan
Enter age:
32
Insert Successfull
BUILD SUCCESSFUL (total time: 18 seconds)
```

UpdatePreparedStatement

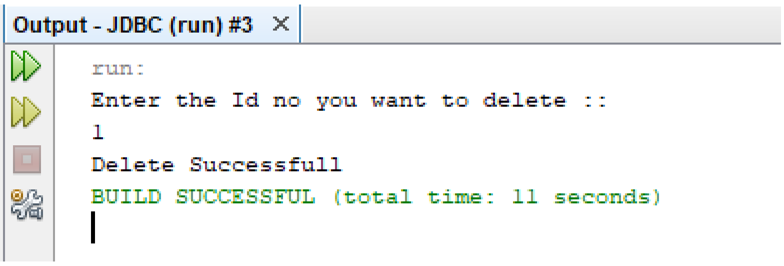
```
package Q4_CRUD_PreparedStatement;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.Scanner;

/**
 *
 * @author user
 */
public class UpdatePreparedStatement {

    public static void main(String[] args) throws SQLException {
        String url = "jdbc:mariadb://localhost:3306/students";
        String username = "root";
        String password = "";
        Connection connection = DriverManager.getConnection(url,
username, password);

        Scanner s = new Scanner(System.in);
        System.out.println("Enter the Id no you want to update ::");
        int id = s.nextInt();
        System.out.println("Enter Name:");
        String name = s.next();
        System.out.println("Enter district:");
        String district = s.next();
        System.out.println("Enter age:");
        int age = s.nextInt();
        String sql = String.format(
                "update studentdata set name='%s',district='%s',"
                + "age='%d' where id = '%d' ", name, district, age,
id
        );
        PreparedStatement statement =
connection.prepareStatement(sql);

        int resultSet = statement.executeUpdate();

        if (resultSet > 0) {
            System.out.println("Update Successfull");
        } else {
            System.out.println("Update Failed (May be the id donot
exist)");
        }
        statement.close();

        connection.close();
    }
}
```



```
run:
Enter the Id no you want to update ::
19
Enter Name:
anup
Enter district:
pokhara
Enter age:
16
Update Successfull
BUILD SUCCESSFUL (total time: 31 seconds)
```

DeletePreparedStatement

```
package Q4_CRUD_PreparedStatement;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.Scanner;

/**
 *
 * @author user
 */
public class DeletePreparedStatement {
    public static void main(String[] args) throws SQLException {
//********************************************** Establishing
Connection to the Database
************************************************
        String url = "jdbc:mariadb://localhost:3306/students";
        String username = "root";
        String password = "";
        Connection connection = DriverManager.getConnection(url,
username, password);

        Scanner s = new Scanner(System.in);
        System.out.println("Enter the Id no you want to delete ::");
        int id = s.nextInt();

        String sql = String.format(
                "delete from studentdata where id = '%d' ", id);

        //********************* Sql Prepared
Statement***********************************
        PreparedStatement statement =
connection.prepareStatement(sql);

        int resultSet = statement.executeUpdate();

        if (resultSet > 0) {
            System.out.println("Delete Successfull");
        } else {
            System.out.println("Delete Failed (May be the id donot
exist)");
        }
        statement.close();

        connection.close();
    }

}
```



```
run:
Enter the Id no you want to delete ::
1
Delete Successfull
BUILD SUCCESSFUL (total time: 11 seconds)
```

5. Implement CRUD operations for student table in Swing. Ask for user input where applicable.

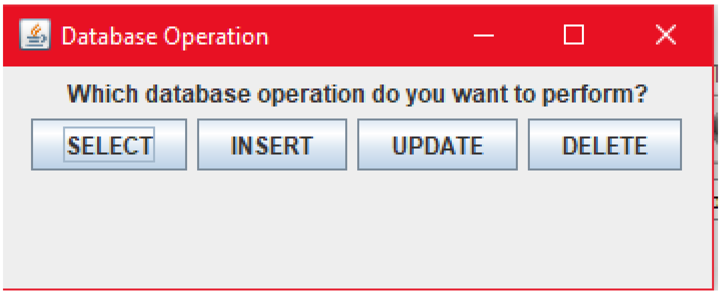
```
package Q5_CRUD_Swing;

import java.awt.FlowLayout;
import java.awt.GridLayout;
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.SQLException;
import java.sql.Statement;
import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JFrame;
import static javax.swing.JFrame.EXIT_ON_CLOSE;
import javax.swing.JLabel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextArea;
import javax.swing.JTextField;

/**
*
* @author user
*/
public class Q5_Swing_CRUD extends JFrame {

public static void main(String[] args) throws SQLException,
NullPointerException {
Q5_Swing_CRUD crud = new Q5_Swing_CRUD();
crud.setVisible(true);
crud.setSize(370, 150);
}
int rows;
Object[][] cells;
JTable table;
ResultSet result;
Statement statement;
Connection connection;
JDialog dialogbox;

public Q5_Swing_CRUD() throws SQLException {

super("Database Operation");
setLayout(new FlowLayout());
dialogbox = new JDialog(Q5_Swing_CRUD.this, "Database");
dialogbox.setLayout(new FlowLayout());
dialogbox.setBounds(100, 70, 500, 100);

JLabel label = new JLabel("Which database operation do you want to
perform?");
add(label);
JButton selectButton = new JButton("SELECT");
add(selectButton);
JButton insertButton = new JButton("INSERT");
add(insertButton);
JButton updateButton = new JButton("UPDATE");
add(updateButton);
JButton deleteButton = new JButton("DELETE");
add(deleteButton);

try {
```

```
String url = "jdbc:mariadb://localhost:3306/students";
String username = "root";
String password = "";
connection = DriverManager.getConnection(url, username, password);
statement = (Statement) connection.createStatement();
} catch (SQLException e) {
System.out.println("Error is" + e);
}

selectButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent ae) {
try {
result = statement.executeQuery("select * from studentdata");
result.last();
rows = result.getRow();
result.first();
} catch (SQLException e) {
System.out.println("Error is" + e);
}
cells = new Object[rows][4];
try {
for (int i = 0; i < rows; i++) {
for (int j = 0; j < 4; j++) {
cells[i][j] = result.getString(j + 1);
}
result.next();
}
} catch (SQLException e) {
System.out.println("Error is" + e);
}
JFrame frame = new JFrame("Selected Database Details!!");
frame.setVisible(true);
frame.setSize(500, 300);
String[] columnNames = {"ID", "Name", "District", "Age"};
table = new JTable(cells, columnNames);
frame.getContentPane().add(new JScrollPane(table));

}
});

insertButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent ae) {
JFrame frame = new JFrame("Insert Into Database!!");
frame.setVisible(true);
frame.setSize(500, 200);
frame.setLayout(new GridLayout(4, 2));

frame.add(new JLabel("Name"));
JTextField nameField = new JTextField(20);
```

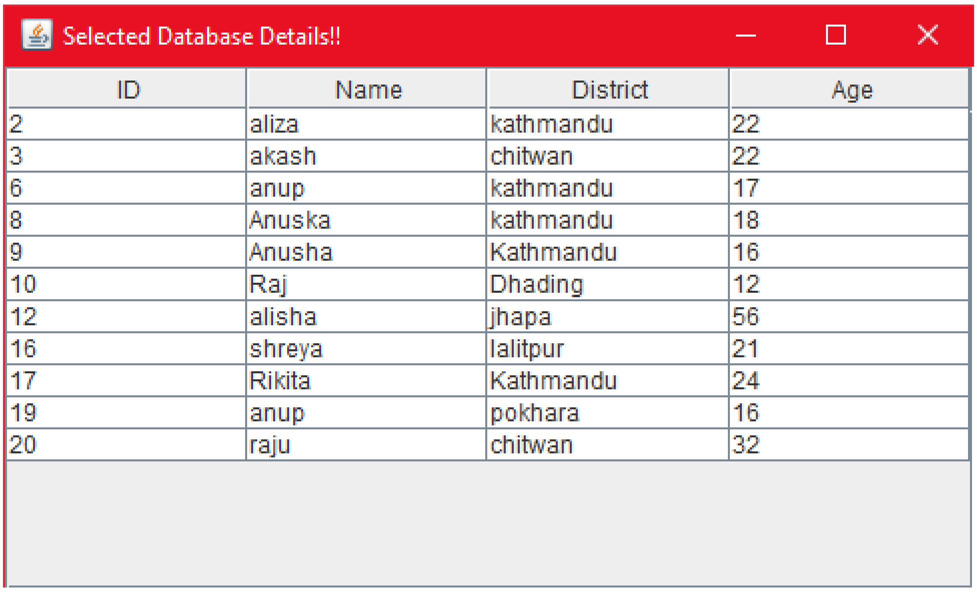
```
frame.add(nameField);

frame.add(new JLabel("District"));
JTextField districtField = new JTextField(20);
frame.add(districtField);

frame.add(new JLabel("Age"));
JTextField ageField = new JTextField(20);
frame.add(ageField);

JButton insert = new JButton("Insert Into Database");
frame.add(insert);

insert.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent ae) {
try {

String name = nameField.getText();
int age = Integer.parseInt(ageField.getText());
String district = districtField.getText();

String sql = String.format(
"insert into studentdata (name, district,age) values ('%s',
'%s','%d')",
name,
district, age
);
int rowsInserted = statement.executeUpdate(sql);
if (rowsInserted > 0) {

JTextArea text = new JTextArea();
text.setText("Insert Succesfull");
dialogbox.add(text);
dialogbox.setVisible(true);

} else {
JTextArea text = new JTextArea();
text.setText("Insert Failed");
dialogbox.add(text);
dialogbox.setVisible(true);
}

statement.close();
connection.close();
} catch (SQLException ex) {
System.out.println("Error is" + ex);
}
}

}
```

```
);
}
});

deleteButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent ae) {
JFrame frame = new JFrame("Delete From Database!!");
frame.setVisible(true);
frame.setSize(500, 200);
frame.setLayout(new GridLayout(4, 2));

frame.add(new JLabel("ID"));
JTextField idField = new JTextField(20);
frame.add(idField);

JButton delete = new JButton("Delete From Database");
frame.add(delete);

delete.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent ae) {
try {

int id = Integer.parseInt(idField.getText());

String sql = String.format(
"delete from studentdata where id='%d'", id
);
int rowsDeleted = statement.executeUpdate(sql);
if (rowsDeleted > 0) {

JTextArea text = new JTextArea();
text.setText("Delete Succesfull");
dialogbox.add(text);
dialogbox.setVisible(true);

} else {
JTextArea text = new JTextArea();
text.setText("Delete Failed");
dialogbox.add(text);
dialogbox.setVisible(true);
}

statement.close();
connection.close();
} catch (SQLException ex) {
System.out.println("Error is" + ex);
}
}
```

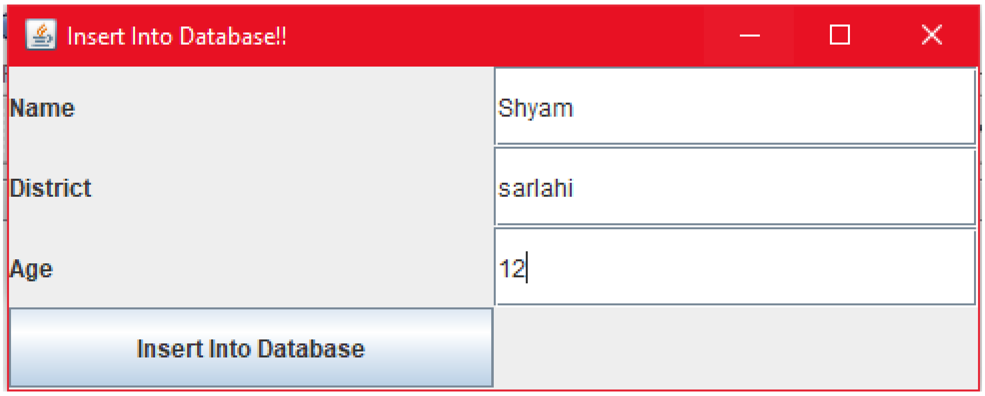
```
}
);
}
});

updateButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent ae) {
JFrame frame = new JFrame("Update Into Database!!");
frame.setVisible(true);
frame.setSize(500, 200);
frame.setLayout(new GridLayout(5, 2));

frame.add(new JLabel("Id"));
JTextField idField = new JTextField(20);
frame.add(idField);

frame.add(new JLabel("Name"));
JTextField nameField = new JTextField(20);
frame.add(nameField);

frame.add(new JLabel("District"));
JTextField districtField = new JTextField(20);
frame.add(districtField);

frame.add(new JLabel("Age"));
JTextField ageField = new JTextField(20);
frame.add(ageField);

JButton update = new JButton("Update Into Database");
frame.add(update);

update.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent ae) {
try {
int id = Integer.parseInt(idField.getText());

String name = nameField.getText();
int age = Integer.parseInt(ageField.getText());
String district = districtField.getText();

String sql = String.format(
"update studentdata set name='%s',district='%s',age='%d' where
id='%d'",
name,
district, age,id
);
ResultSet rs = statement.executeQuery(sql);
statement.close();
connection.close();
} catch (SQLException ex) {
System.out.println("Error is" + ex);
}
}

}
);
}
});


pack();
setDefaultCloseOperation(EXIT_ON_CLOSE);

}
}
```



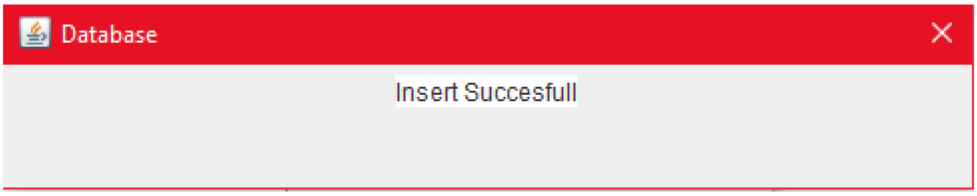
Which database operation do you want to perform?



ID	Name	District	Age
2	aliza	kathmandu	22
3	akash	chitwan	22
6	anup	kathmandu	17
8	Anuska	kathmandu	18
9	Anusha	Kathmandu	16
10	Raj	Dhading	12
12	alisha	jhapa	56
16	shreya	lalitpur	21
17	Rikita	Kathmandu	24
19	anup	pokhara	16
20	raju	chitwan	32



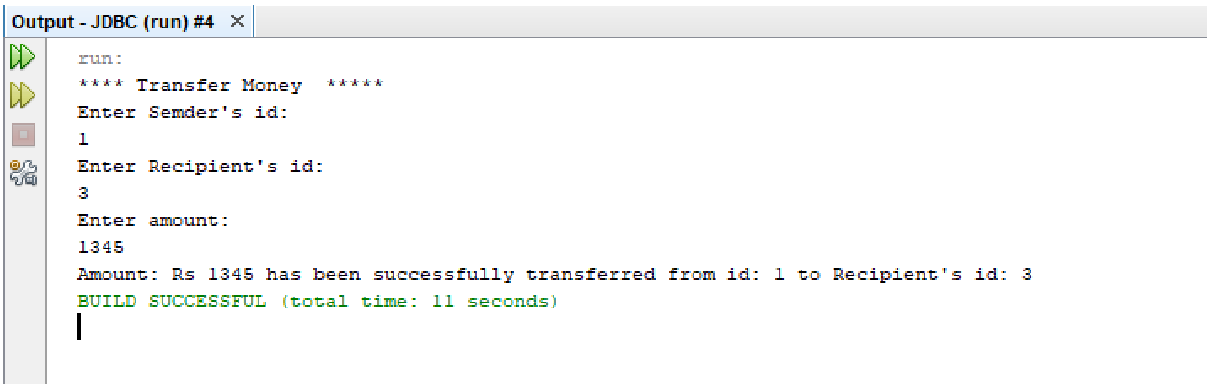
Name	Shyam
District	sarlahi
Age	12



Insert Succesfull

6. Implement account balance transfer operation (use transactions). Ask for user input where applicable.

```
package Q6_SQL_Transaction;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;

/**
*
* @author user
*/
public class Q6_BalanceTransfer {

public static void main(String[] args) throws SQLException {
String url = "jdbc:mariadb://localhost:3306/bank_transactions";
String username = "root";
String password = "";
Connection connection = DriverManager.getConnection(url,
username, password);
Statement statement = (Statement) connection.createStatement();
connection.setAutoCommit(false);

Scanner in = new Scanner(System.in);

System.out.println("**** Transfer Money  *****");
System.out.println("Enter Semder's id: ");
int senderId = in.nextInt();

System.out.println("Enter Recipient's id: ");
int recipientId = in.nextInt();

System.out.println("Enter amount: ");
int amount = in.nextInt();

try {

String sqlsenderupdate = String.format(
"update account_detail set Balance=Balance-'%d' where Id='%d'",
amount, senderId
);

String sqlreceiverupdate = String.format(
"update account_detail set Balance=Balance+'%d' where Id='%d'",
amount, recipientId
);

ResultSet senderupdate = statement.executeQuery(sqlsenderupdate);
ResultSet receiverupdate =
statement.executeQuery(sqlreceiverupdate);
connection.commit();

System.out.println("Amount: Rs " + amount +" has been
successfully transferred from id: " + senderId+
" to Recipient's id: "+ recipientId);

}
catch(SQLException e){
connection.rollback();
System.out.println("Transfer Failed!!!Try Again.....");
}

}
}
```



```
run:
**** Transfer Money  *****
Enter Semder's id:
1
Enter Recipient's id:
3
Enter amount:
1345
Amount: Rs 1345 has been successfully transferred from id: 1 to Recipient's id: 3
BUILD SUCCESSFUL (total time: 11 seconds)
```