

**Java**

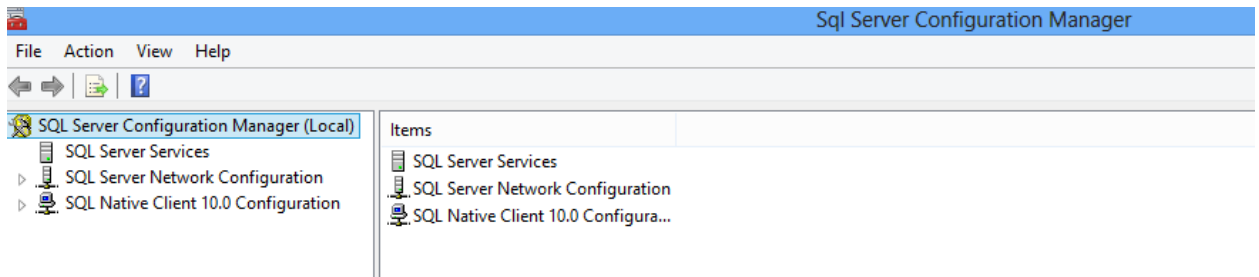
**To**

**SQL Server Database Connectivity**

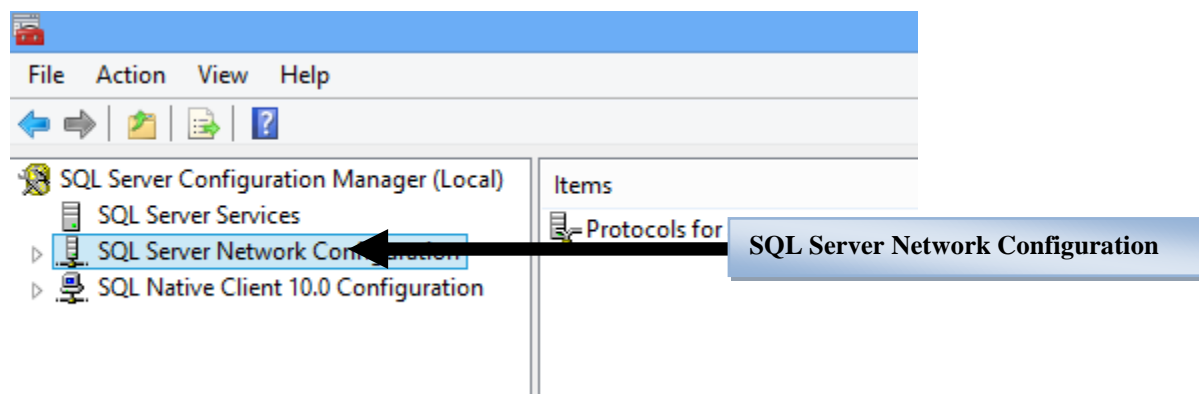
**Today, we will cover the followings**

- ✓ Creating a SQL Server Database
- ✓ Creating a table in SQL Server Database
- ✓ Configuring SQL Server with Java
- ✓ Designing a JFrameForm for adding Data to SQL Server Database

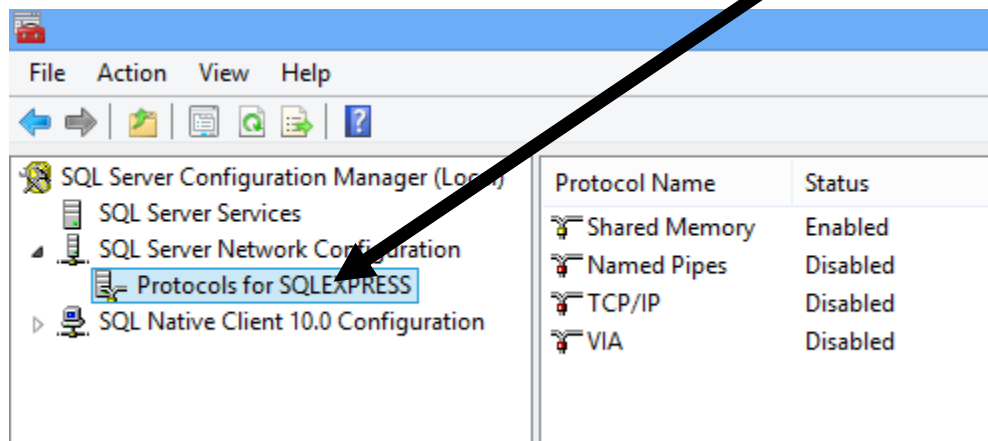
1. Look for and Open **SQL Server Configuration Manager** as shown in the figure below



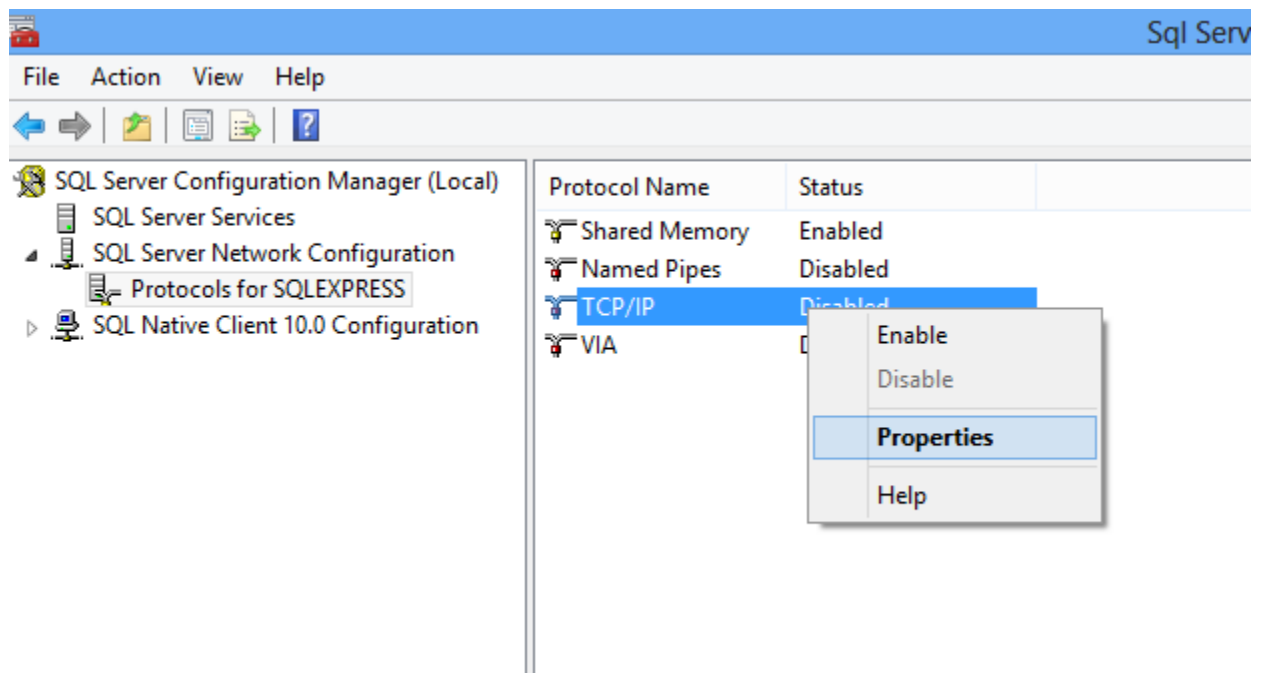
2. Next, Select and expand **SQL Server Network Configuration** as shown in the figure below



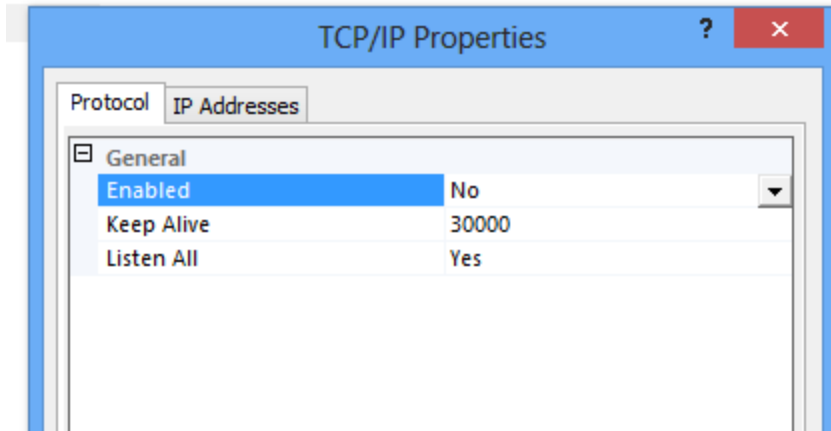
- Expand **SQL Server Network Configuration** and select **Protocols for SQLEXPRESS**



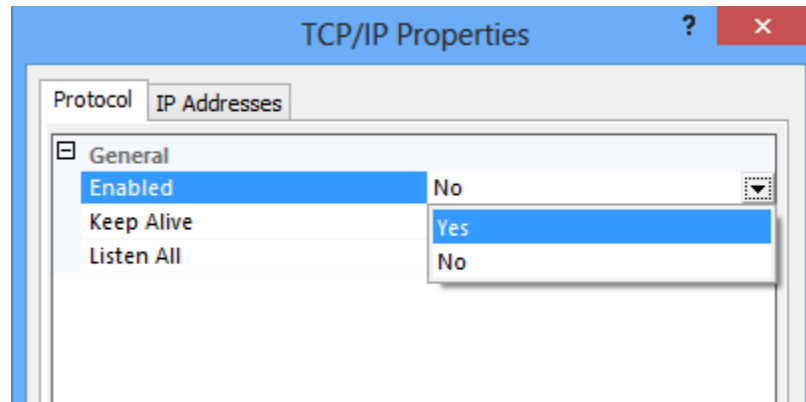
- On the Right hand pane, Select the **TCP/IP**, **Right Click on TCP/IP**



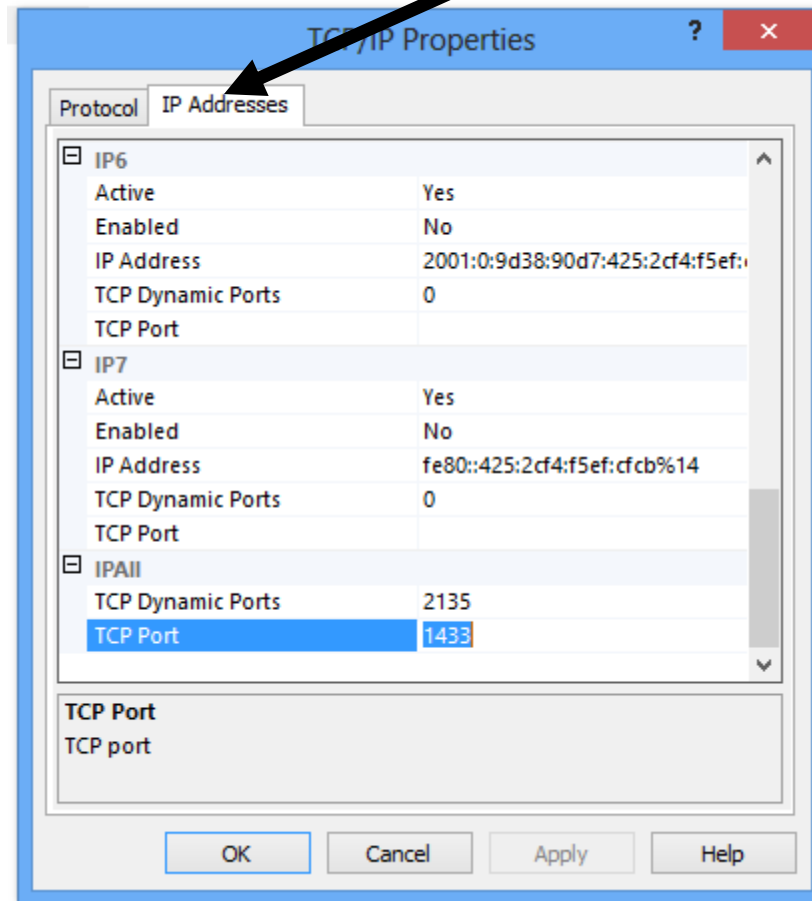
- Click on Properties, this will open the following Window



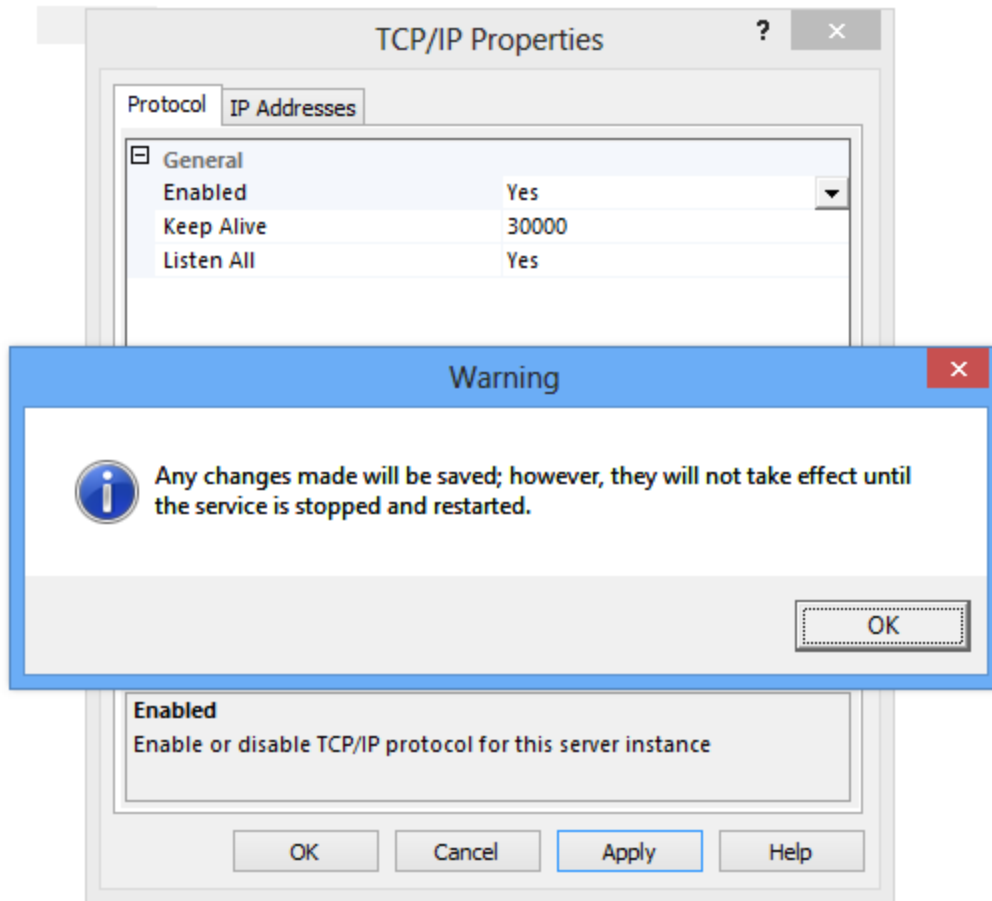
- Set the **Enabled** property to **YES** from dropdown



7. Next, Click on the second Tab in the same Window titled “**IP Addresses**”

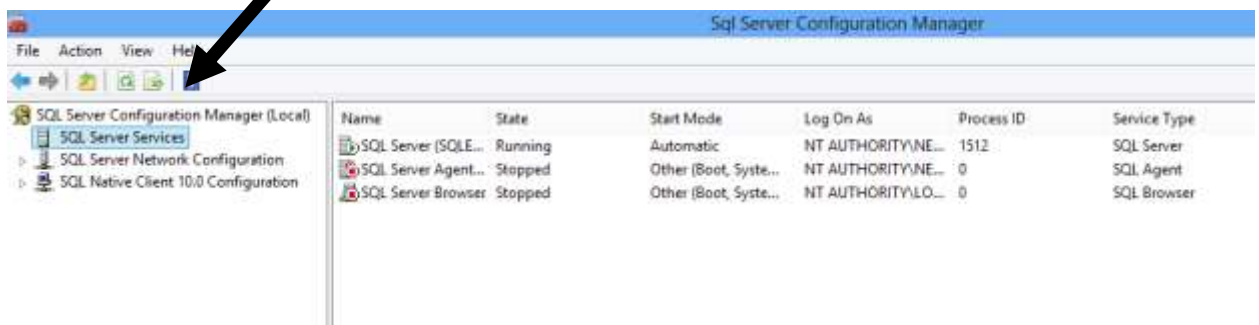


8. In the **IPAll**, enter the value **1433** in the **TCP Port** Click the **OK Button**, this will open the Window as shown

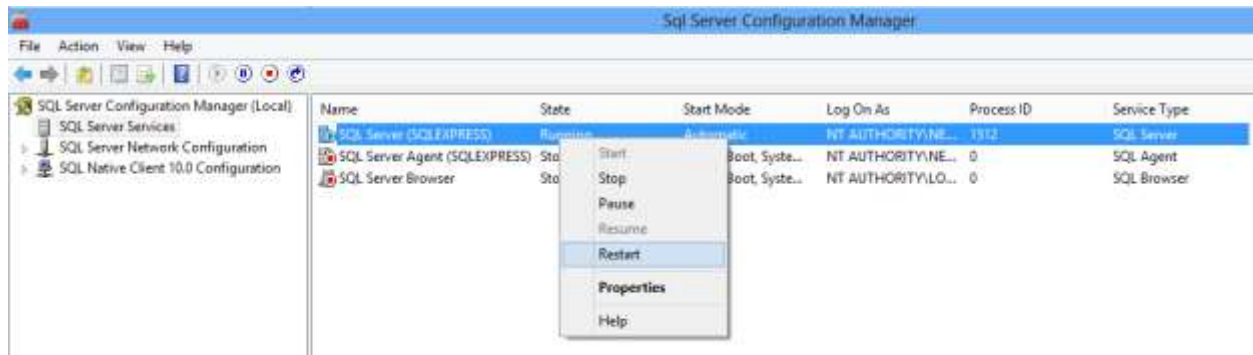


9. Press the **OK Button**

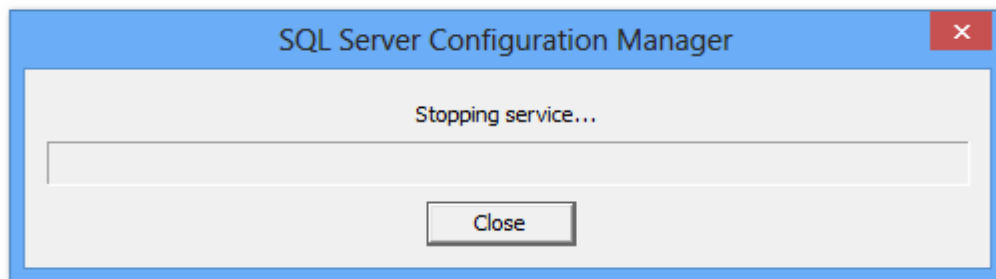
10. These changes will reflect when you **Restart the SQL Server**. Therefore, to restart the SQL Server, expand **SQL Server Services node**, located above the SQL Server Network Configuration.



11. On the Right pane, Right Click The **SQL Server** and Restart it, as shown in the next figure



12. Restart of SQL Server is in process

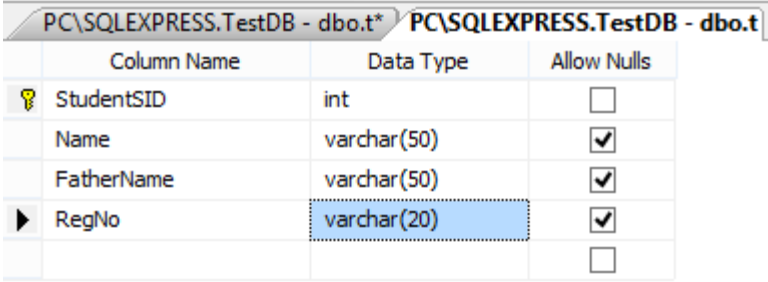




## Database Creation Steps

1. Next, Suppose we have created the following table in the database name (We have performed this Task in detail during last week)

**Table 1 Database, database Table details**

<b>Database Name:</b>	UIITStudentsForum
<b>Table Name :</b>	tblStudentsDetails
<b>Table Structure</b>	
<b>Primary Key:</b>	StudentSID (Auto Generated)

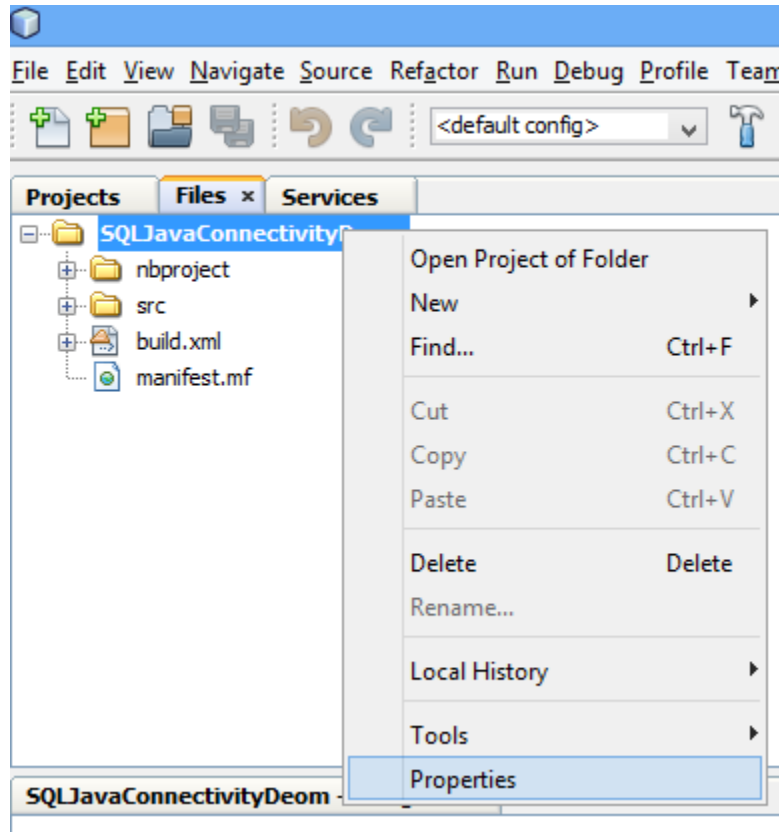
2. To connect Java to MS SQL Server, you will need **Microsoft JDBC Driver 4.1 for SQL Server**, this can be downloaded from Microsoft Official Web site, the Link is given below

**<https://www.microsoft.com/en-pk/download/details.aspx?id=11774>**

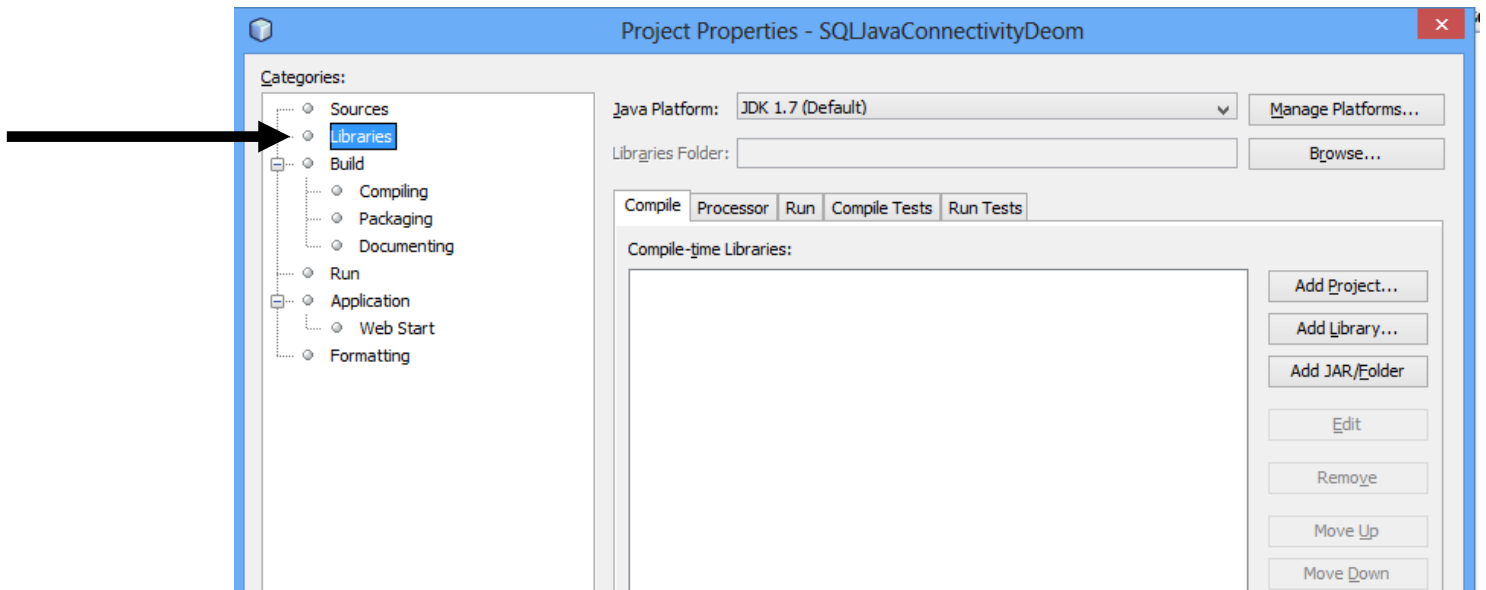
3. Download this driver to your Computer

## Working in Netbeans

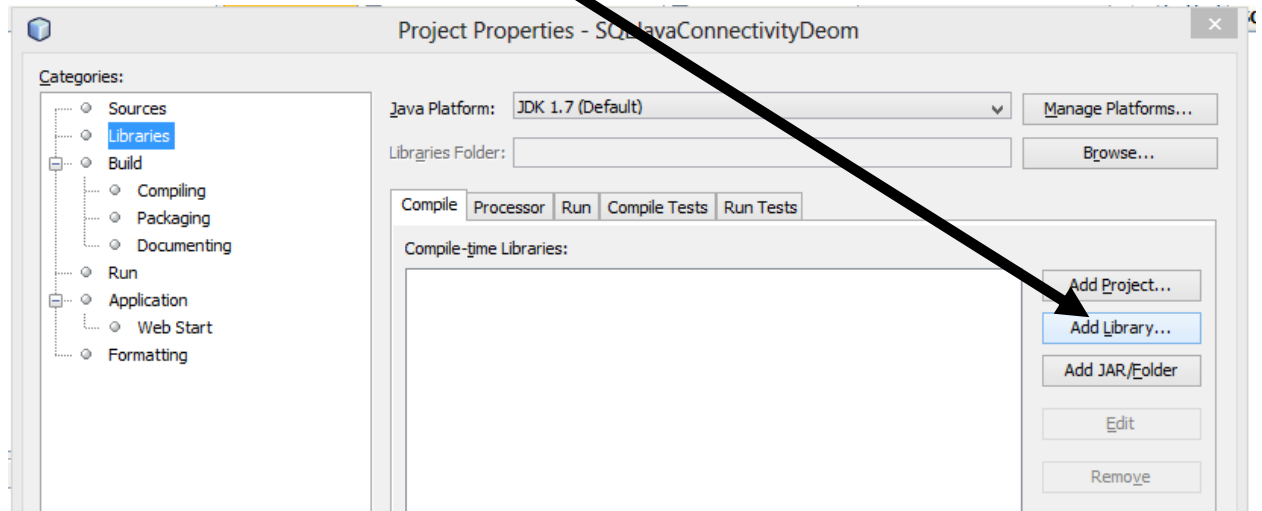
1. Next, follow the following steps to add the **sqljdbc41.jar** to your Java Project Library, that you have downloaded from Microsoft web site in the previous step
2. Right Click the Project and then Click the Properties



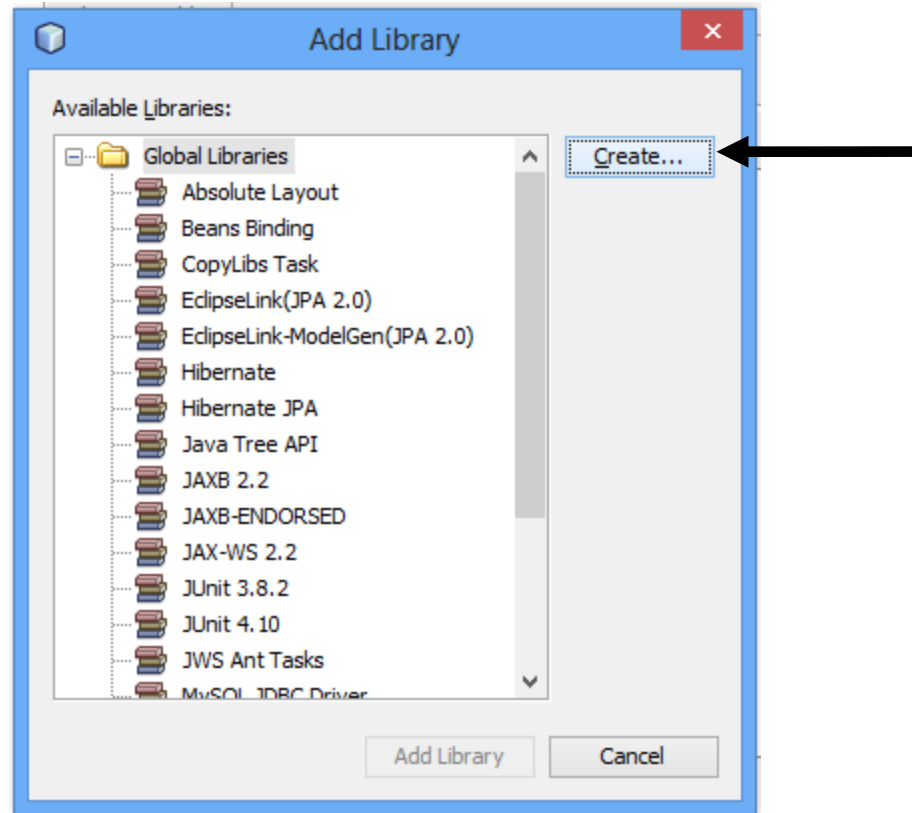
3. Select the Libraries



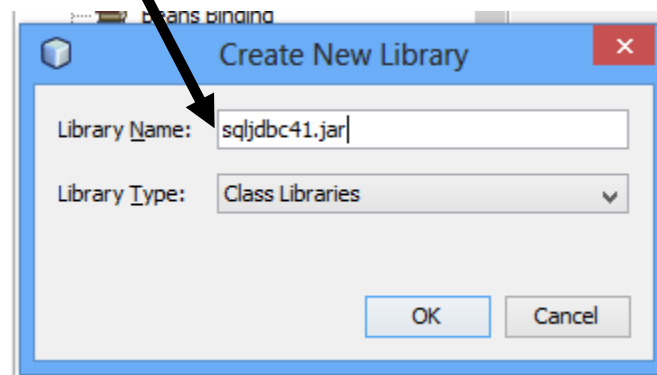
4. Click the Add Library



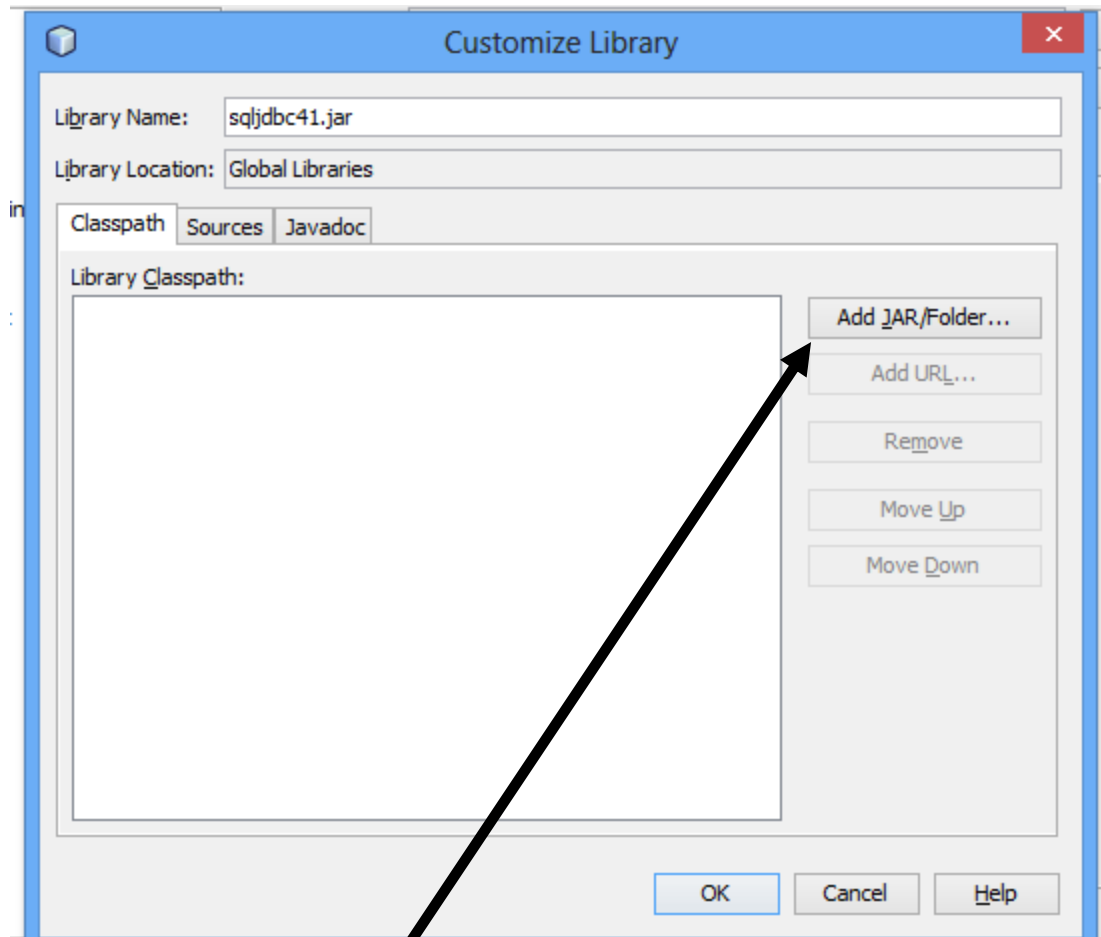
5. In the next window click **CREATE**



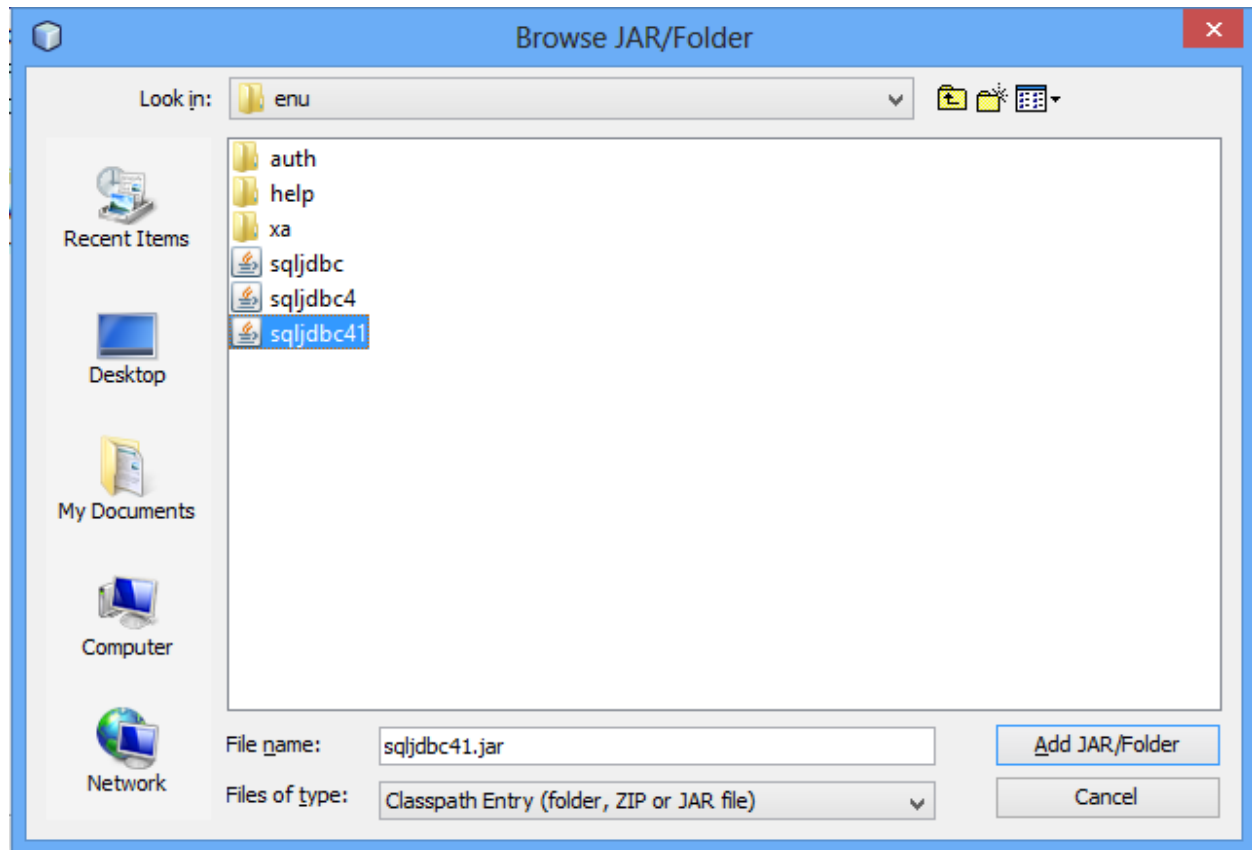
6. Add the **Library Name**



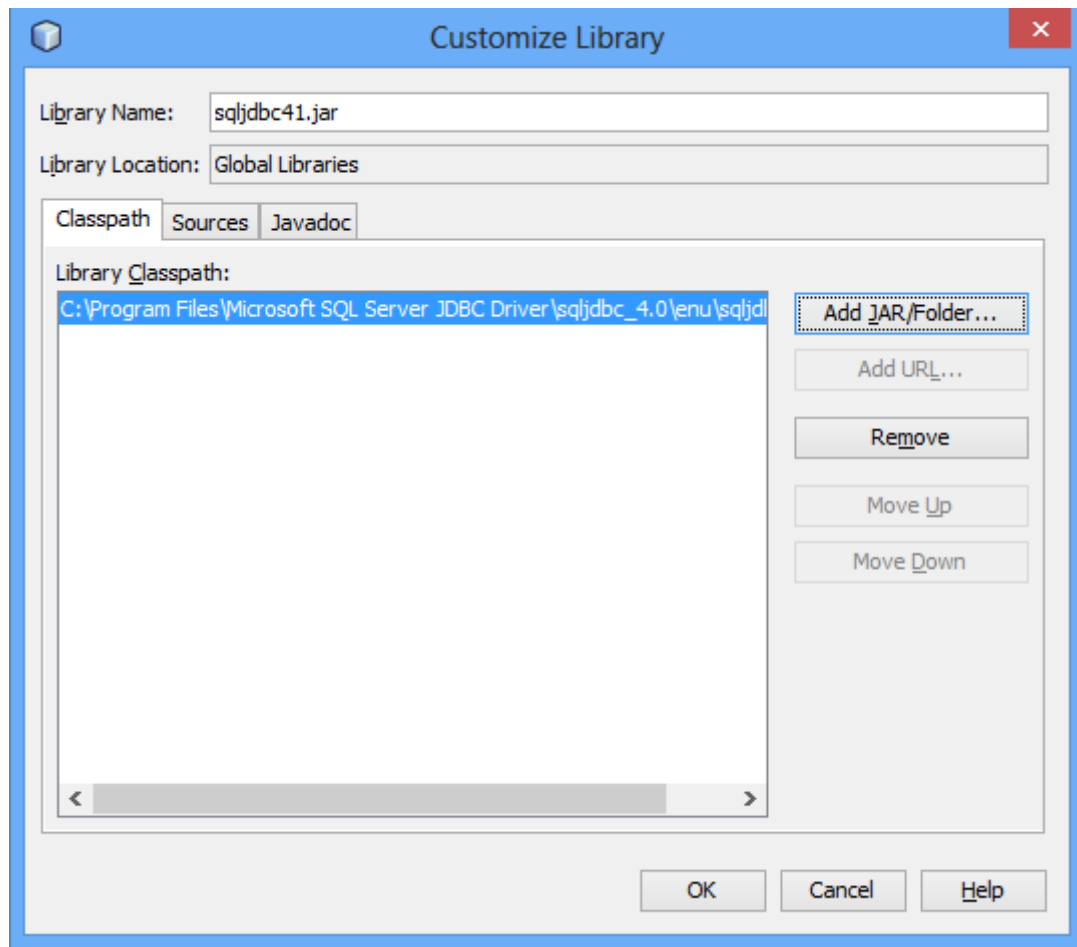
7. Press the OK button, this will open new Window



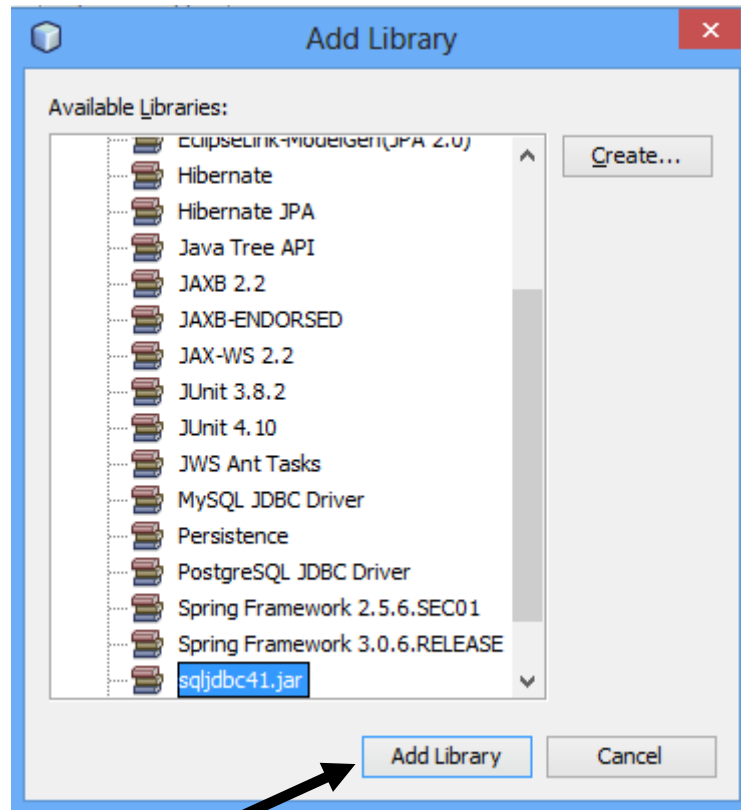
8. Press the Add Jar/Folder button



9. Locate the **sqljdbc41.jar** and press the **Add Jar/Folder** button

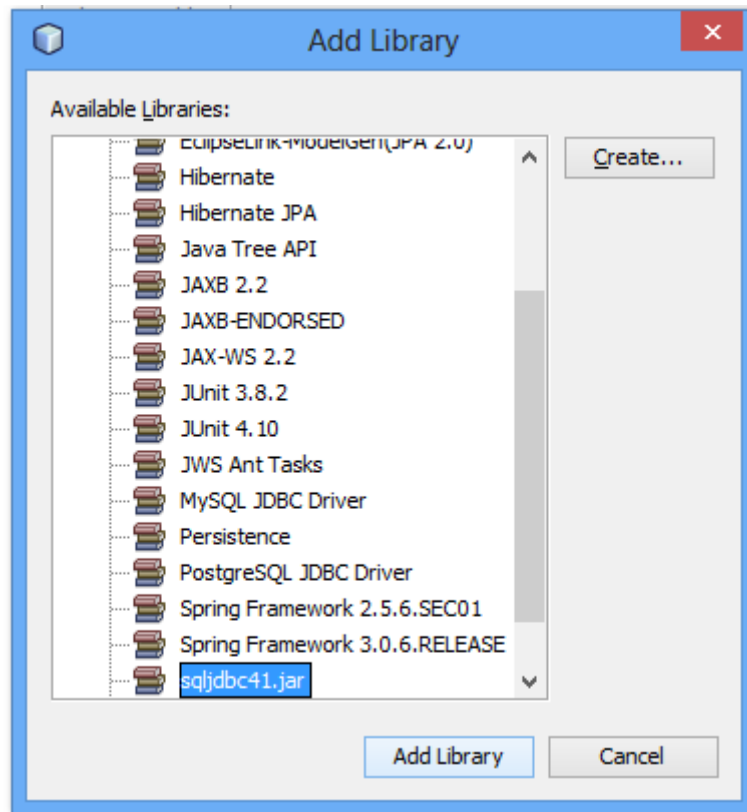


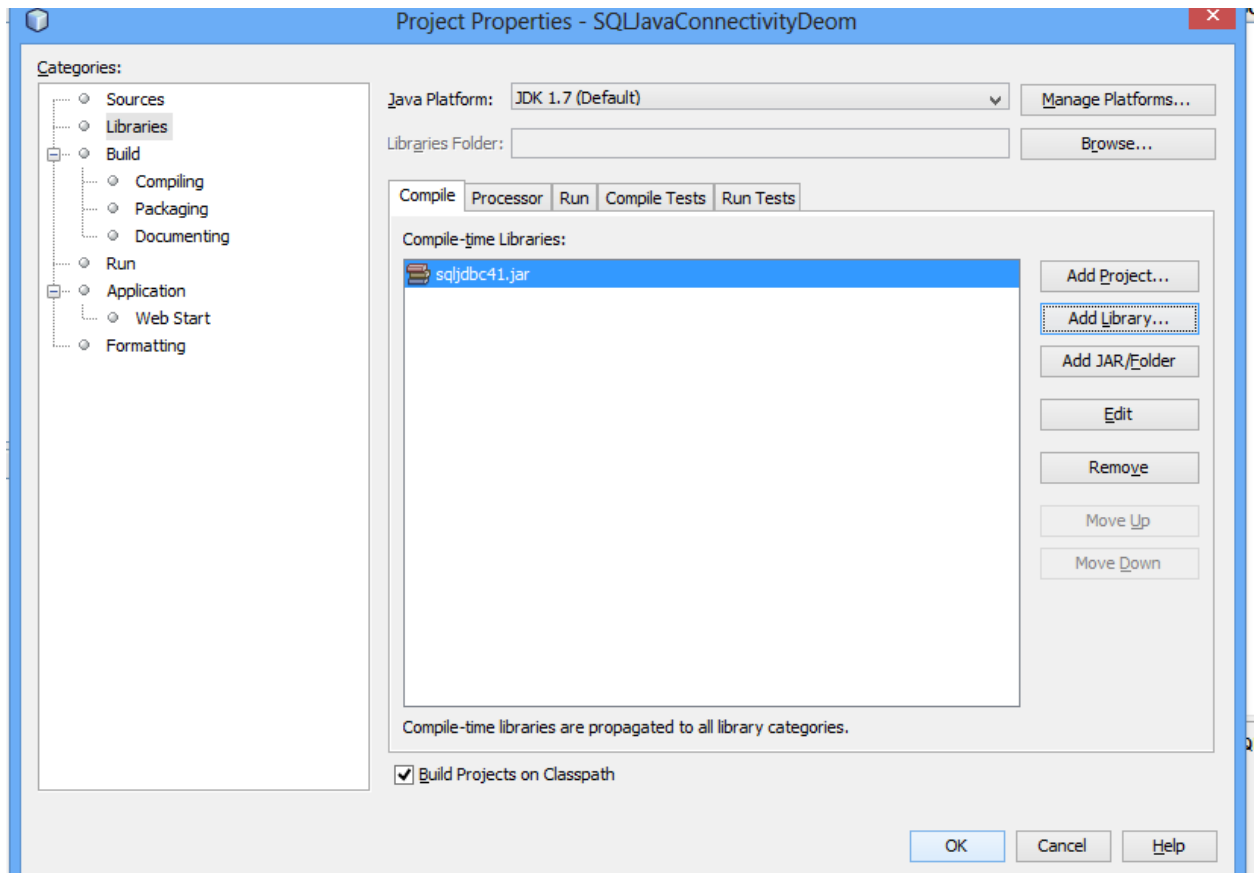
10. Press the OK button, this will add the Library



11. Now, press the **Add Library** button in the above figure, this will add the selected Jar file to your project, which will be further used to connect SQL Server Database to Java jFrameform.







12. Finally, press the OK button

13. Start Add a new Project in Netbeans and Rename the Project to

**SQLJavaConnectivityDemo**

14. Add a JFrameform and next design the form for Java to SQL connectivity as shown

The image shows a NetBeans IDE window titled "UIIT Students Registration Form". The window contains a Java Swing form with three labels: "Student Name", "Registration No", and "Father Name". Each label is followed by a white text input field. At the bottom of the form, there are two buttons: "Save Record" and "New Record".

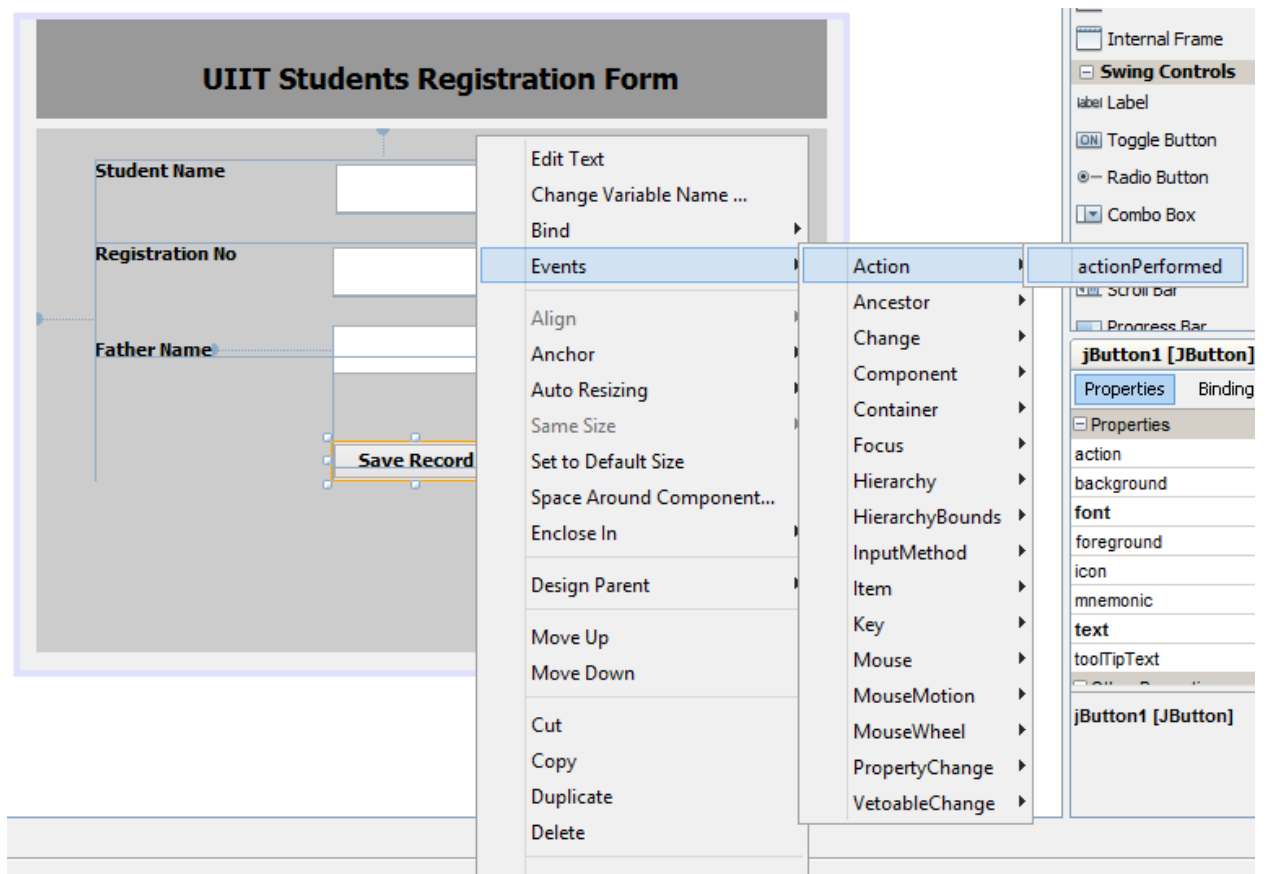
15. Name of each control is given in the table below and follow naming conventions like this for all of your future projects, as giving meaningful names to each control is considered the sign of a good programmer.

**Table 2. Naming Conventions for Various Netbeans Controls**

S#	Control	Name	Label/Title
1	jLabel1	lblName	Student Name
2	jLabel2	lblRegNo	Registration Number
3	jLabel3	lblFatherName	Father Name
4	(jTextField1	txtStdName	
5	(jTextField2	txtRegNo	
6	(jTextField3	txtFatherName	

7	jButton1	btnSave	Save Record
8	jButton2	btnNEW	New Record

16. Now, open the designed form. Right click the Save Record button for Event



17. Add the following built-in java package

```

11 import java.sql.*;
12
13 public class NewJFrame extends javax.swing.JFrame {
14
15     /**
16      * Creates new form NewJFrame
17      */
18     public NewJFrame() {
19         initComponents();
20     }
21
22     /**
23      * This method is called from within the constructor to initialize the form.
24      * WARNING: Do NOT modify this code. The content of this method is always
25      * regenerated by the Form Editor.

```

### java.sql.\*

jav.sql is a package which contains useful classes, helping your java project to Communicate with a SQL (Structured Query Language) database.

18. Next, add the following code in the actionPerformed Event of **Save Record** button.

These lines specifies the Database Name, User Id and Password and other details regarding the database to which data will be saved.

The line

**Connection myCon = DriverManager.getConnection(url);**

Instantiate the connection class and then establishes the connection of Java to SQL

Server. The very next line output the Connection Successful message if the connection is established successfully...

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");

        String url = "jdbc:sqlserver://localhost:1433;databaseName=UIITStudentsForum;user=sa;password=rehman";

        Connection myCon = DriverManager.getConnection(url);
        System.out.println("Connection to Database opened sucessfully.....");
    }
    catch (Exception e){
        System.out.println(e.toString());
    }
}

```

### Plain code text

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
    try {
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
        String url =
            "jdbc:sqlserver://localhost:1433;databaseName=UIITStudentsForum;user=sa;password=r
            ehman";
        Connection myCon = DriverManager.getConnection(url);
    }
    catch (Exception e){
        System.out.println(e.toString());
    }
}
```

19. Compile and Run the JFrame to check whether the connection has been established successfully or not.

```
run:
Connection to Database opened sucessfully.....
BUILD SUCCESSFUL (total time: 1 second)
```

**Congratulations! We have successfully established the Java-SQL Server database connection.**

# Using Insert Into Statement to Insert new Record to SQL

## Server database table

### SQL Insert Into Statement:

To add a new record into SQL Server table, SQL Insert Into Statement is used. The syntax for Insert Into Statement is as follow

Insert Into **TableName** (Col11, Col2,,,,,,Clon)

Values

(Val1, Val2,,,,,,, Val,)

1. Type in the following Code in the Click Event of the Save Record Button

```
167 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
168     try {  
169         Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");  
170  
171         String url = "jdbc:sqlserver://localhost:1433;databaseName=UIIISStudentsForum;user=sa;password=rehman";  
172  
173  
174         Connection myCon = DriverManager.getConnection(url);  
175         // create a Statement from the connection  
176         Statement statement = myCon.createStatement();  
177  
178         String stdName, regNo, fatherName;  
179  
180         stdName = txtStdName.getText();  
181  
182         regNo = txtRegNo.getText();  
183         fatherName = txtFatherName.getText();  
184  
185         String sqlInsert = "INSERT INTO sblStudentsDetails VALUES ('" + stdName + "','" + regNo + "','" + fatherName + "')";  
186  
187         // insert the data  
188         statement.executeUpdate(sqlInsert);  
189     }  
190     catch (Exception e) {  
191         System.out.println(e.toString());  
192     }  
193 }  
194
```

2. Code in Plain text format is given by

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try {
```

```

Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");

String url =

"jdbc:sqlserver://localhost:1433;databaseName=UIITStudentsForum;user=sa;pass
word=rehman";

Connection myCon = DriverManager.getConnection(url); ← L1

// create a Statement from the connection

Statement statement = myCon.createStatement(); ← L2

String stdName, regNo, fatherName;

stdName = txtStdName.getText();

regNo = txtRegNo.getText();

fatherName = txtFatherName.getText();

String sqlInsert = "INSERT INTO tblStudentsDetails VALUES ('" + stdName
+ "','" + regNo + "','" + fatherName + "')";

// insert the data

statement.executeUpdate(sqlInsert); ← L3

}

catch (Exception e){

System.out.println(e.toString());

}

}

```

**L1** The line

**Statement statement = myCon.createStatement();**



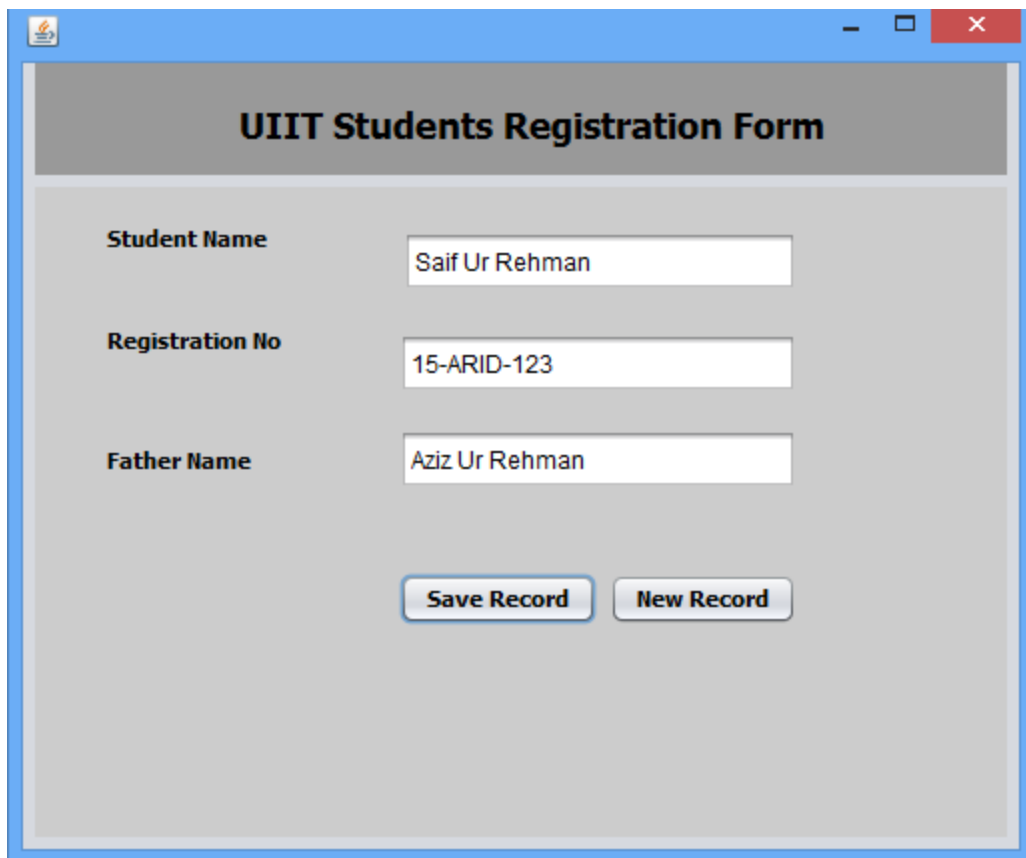
Instantiate the Statement class for creating the Statement, which will be further used for inserting data to SQL Server database.

**L2** This line declares three variables to get the data from three JTextfield and these will be used further in **INSERT INTO** statement.

**L3** This line

**statement.executeUpdate(sqlInsert);**

Will execute the SqlInsert statement to add new data record to database table

The image shows a Java Swing window titled "UIIT Students Registration Form". The window has a light gray background and a blue title bar. Inside the window, there are three text input fields arranged vertically. The first field is labeled "Student Name" and contains the text "Saif Ur Rehman". The second field is labeled "Registration No" and contains the text "15-ARID-123". The third field is labeled "Father Name" and contains the text "Aziz Ur Rehman". Below the input fields, there are two buttons: "Save Record" and "New Record". The "Save Record" button is highlighted with a blue border. The window has standard Windows-style window controls (minimize, maximize, close) in the top right corner.

3. Press **Save Record** button
4. Further, verify the record in the database

PC\SQLEXPRESS.UI...IStudentsDetails		PC\SQLEXPRESS.UI...IStudentsDetails		
	StudentSID	Name	FatherName	RegNo
▶	7	Saif Ur Rehman	15-ARID-123	Aziz Ur Rehman
*	NULL	NULL	NULL	NULL

**That's All! We have successfully done today by**

- ✓ Creating a SQL Server to Java database Connection
- ✓ Designing Database and Table
- ✓ Coding in Java to add Data
- ✓ Compiling and Running the Application

**Saif Ur Rehman**

Assistant Professor

UIIT, PMAS Arid Agriculture University, Rawalpindi