Selenium Webdriver is limited to [Testing](https://www.guru99.com/software-testing.html) your applications using Browser. To use Selenium Webdriver for Database Verification you need to use the JDBC ("Java Database Connectivity").

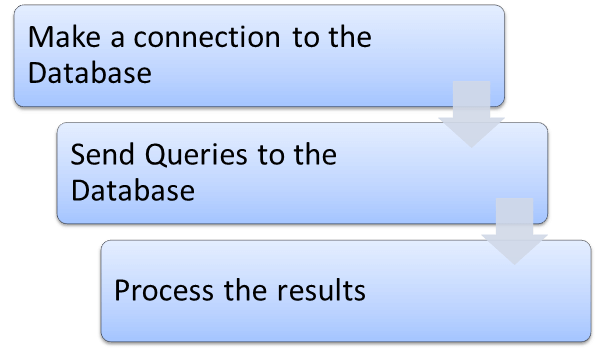
JDBC (Java Database Connectivity) is a [SQL](https://www.guru99.com/sql.html) level API that allows you to execute SQL statements. It is responsible for the connectivity between the [Java](https://www.guru99.com/java-tutorial.html) Programming language and a wide range of databases. The JDBC API provides the following classes and interfaces

* Driver Manager
* Driver
* Connection
* Statement
* ResultSet
* SQLException

In this tutorial, you will learn

* [Make a connection to the Database](https://www.guru99.com/database-testing-using-selenium-step-by-step-guide.html#1)
* [Send Queries to the Database](https://www.guru99.com/database-testing-using-selenium-step-by-step-guide.html#2)
* [Process the results](https://www.guru99.com/database-testing-using-selenium-step-by-step-guide.html#3)
* [Example of Database Testing with Selenium](https://www.guru99.com/database-testing-using-selenium-step-by-step-guide.html#4)

In order to test your Database using Selenium, you need to observe the following 3 steps

[](https://www.guru99.com/images/6-2015/052615_1013_DatabaseTes1.png)

**1) Make a connection to the Database**

In order to make a connection to the database the syntax is

DriverManager.getConnection(URL, "userid", "password" )

Here,

* Userid is the username configured in the database
* Password of the configured user
* URL is of format jdbc:< dbtype>://ipaddress:portnumber/db\_name"
* <dbtype>- The driver for the database you are trying to connect. To connect to oracle database this value will be "oracle"

For connecting to database with name "emp" in MYSQL URL will bejdbc:mysql://localhost:3036/emp

And the code to create connection looks like

Connection con = DriverManager.getConnection(dbUrl,username,password);

You also need to load the JDBC Driver using the code

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

**2) Send Queries to the Database**

Once connection is made, you need to execute queries.

You can use the Statement Object to send queries.

Statement stmt = con.createStatement();

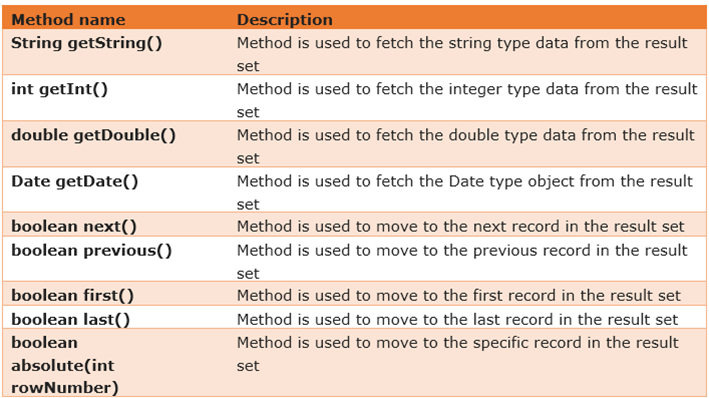
Once the statement object is created use the executeQuery method to execute the SQL queries

stmt.executeQuery(select \* from employee;);

**3) Process the results**

Results from the executed query are stored in the ResultSet Object.

Java provides loads of advance methods to process the results. Few of the methods are listed below

[](https://www.guru99.com/images/6-2015/052615_1013_DatabaseTes2.png)

**Example of Database Testing with Selenium**

**Step 1)** Install [MySQL Server](http://dev.mysql.com/downloads/mysql/) and [MySQL Workbench](http://dev.mysql.com/downloads/workbench/)

Check out the complete guide to Mysql & Mysql Workbench [here](https://www.guru99.com/sql.html)

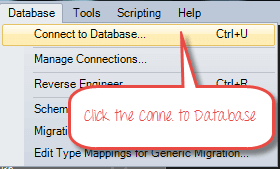
While installing MySQL Server, please note the database

* Username
* Password
* Port Number

It will be required in further steps.

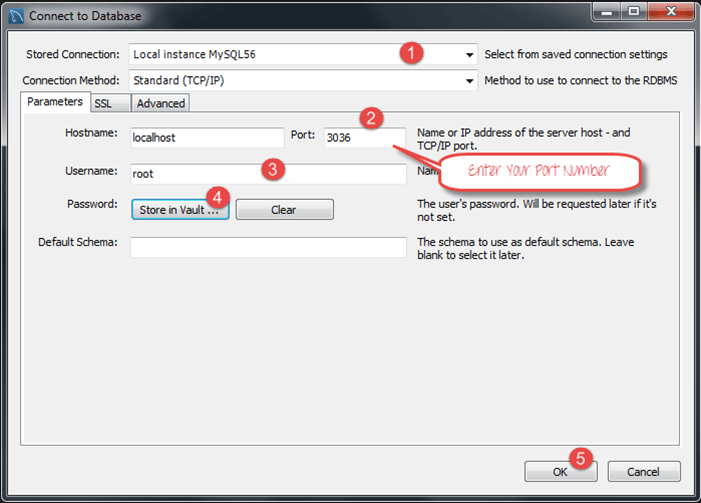
MySQL Workbench makes it easy to administer the database without the need to code SQL. Though, you can also use the MySQL Terminal to interact with the database.

**Step 2)** In MySQL WorkBench, connect to your MySQL Server

[](https://www.guru99.com/images/6-2015/052615_1013_DatabaseTes3.png)

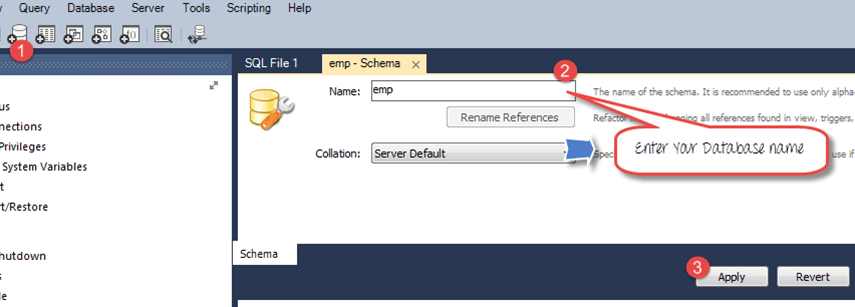
In the next screen,

1. Select Local Instance of MySQL
2. Enter Port Number
3. Enter Username
4. Enter Password
5. Click OK

[](https://www.guru99.com/images/6-2015/052615_1013_DatabaseTes4.png)

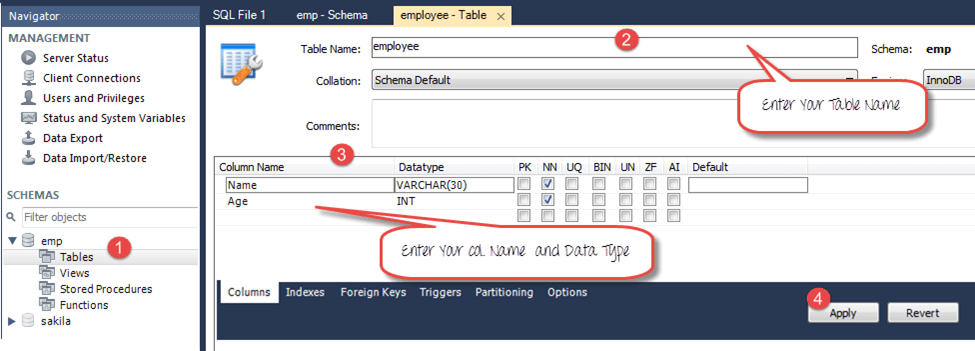
**Step 3)** To Create Database,

1. Click create Schema Button
2. Enter Name of Schema/Database
3. Click Apply

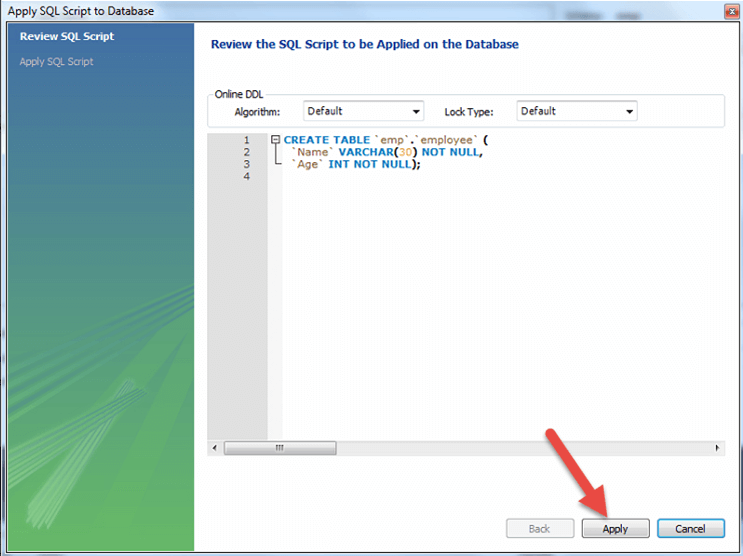
[](https://www.guru99.com/images/6-2015/052615_1013_DatabaseTes5.png)

**Step 4)** In the navigator menu,

1. Click on Tables, beneath the emp database
2. Enter Table name as employee
3. Enter Fields as Name and Age
4. Click Apply

[](https://www.guru99.com/images/6-2015/052615_1013_DatabaseTes6.png)

You will see the following pop-up. Click Apply

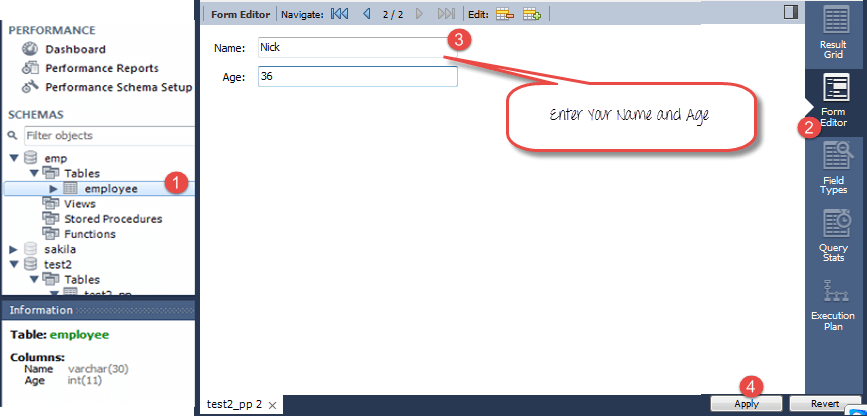
[](https://www.guru99.com/images/6-2015/052615_1013_DatabaseTes7.png)

**Step 5)** We will create following data

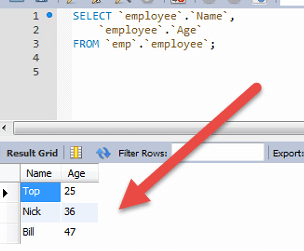
|  |  |
| --- | --- |
| **Name** | **Age** |
| Top | 25 |
| Nick | 36 |
| Bill | 47 |

To create data into the Table

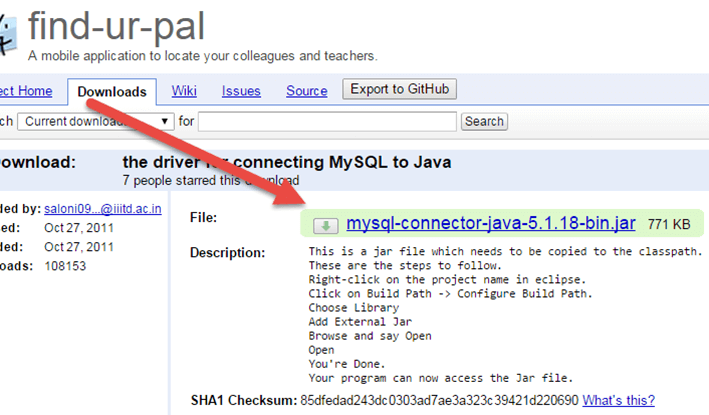
1. In navigator, select the employee table
2. In right pane, click Form Editor
3. Enter Name and Age
4. Click Apply

[](https://www.guru99.com/images/6-2015/052615_1013_DatabaseTes8.png)

Repeat the process until all data is created

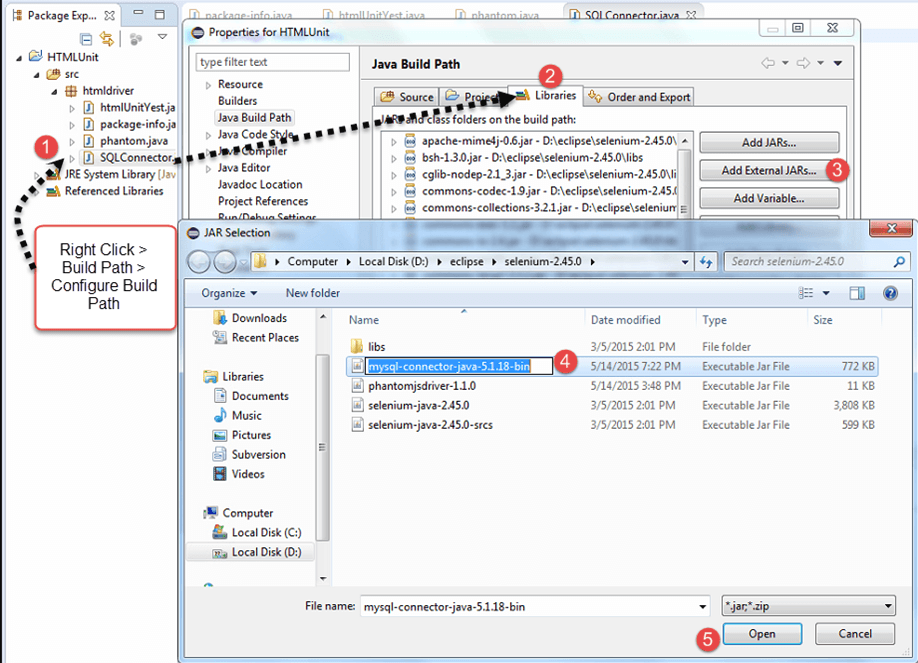
[](https://www.guru99.com/images/6-2015/052615_1013_DatabaseTes9.png)

**Step 6)** Download the MySQL JDBC connector [here](https://code.google.com/p/find-ur-pal/downloads/detail?name=mysql-connector-java-5.1.18-bin.jar&)

[](https://www.guru99.com/images/6-2015/052615_1013_DatabaseTes10.png)

**Step 7)** Add the downloaded Jar to your Project

1. Right click on your Java File. Then click on Build Pathà Configure build path
2. Select the libraries
3. Click on add external JARs
4. You can see MySQL connector java in your library
5. Click on open to add it to the project

[](https://www.guru99.com/images/6-2015/052615_1013_DatabaseTes11.png)

**Step 8)** Copy the following code into the editor

Package htmldriver;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

import java.sql.DriverManager;

import java.sql.SQLException;

public class SQLConnector {

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

//Connection URL Syntax: "jdbc:mysql://ipaddress:portnumber/db\_name"

String dbUrl = "jdbc:mysql://localhost:3036/emp";

//Database Username

String username = "root";

//Database Password

String password = "guru99";

//Query to Execute

String query = "select \* from employee;";

//Load mysql jdbc driver

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

//Create Connection to DB

Connection con = DriverManager.getConnection(dbUrl,username,password);

//Create Statement Object

Statement stmt = con.createStatement();

// Execute the SQL Query. Store results in ResultSet

ResultSet rs= stmt.executeQuery(query);

// While Loop to iterate through all data and print results

while (rs.next()){

String myName = rs.getString(1);

String myAge = rs.getString(2);

System. out.println(myName+" "+myAge);

}

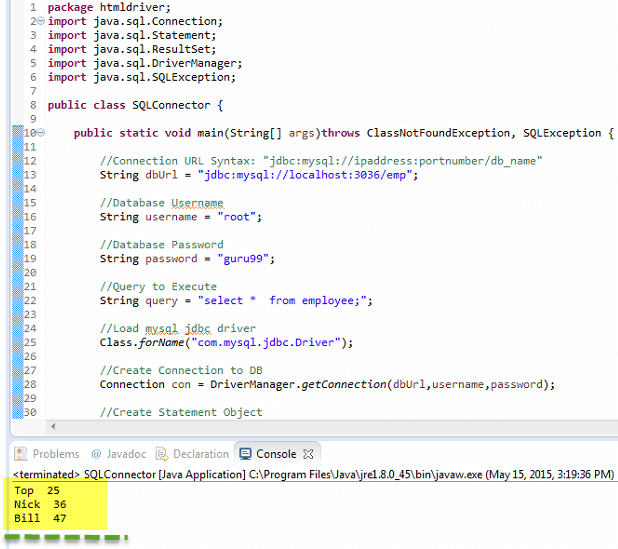
// closing DB Connection

con.close();

}

}

**Step 8)** Execute the code, and check the output

[](https://www.guru99.com/images/6-2015/052615_1013_DatabaseTes12.png)

**Summary of Steps for Selenium Database Testing**

**Step 1)** Make a connection to the Database using method.

DriverManager.getConnection(URL, "userid", "password")

**Step 2)** Create Query to the Database using the Statement Object.

Statement stmt = con.createStatement();

**Step 3)** Send the query to database using execute query and store the results in the ResultSet object.

ResultSet rs = stmt.executeQuery(select \* from employee;);

Java provides lots of built-in methods to process the> SQL Output using the ResultSet Object