

Database Testing using Selenium: Step by Step Guide

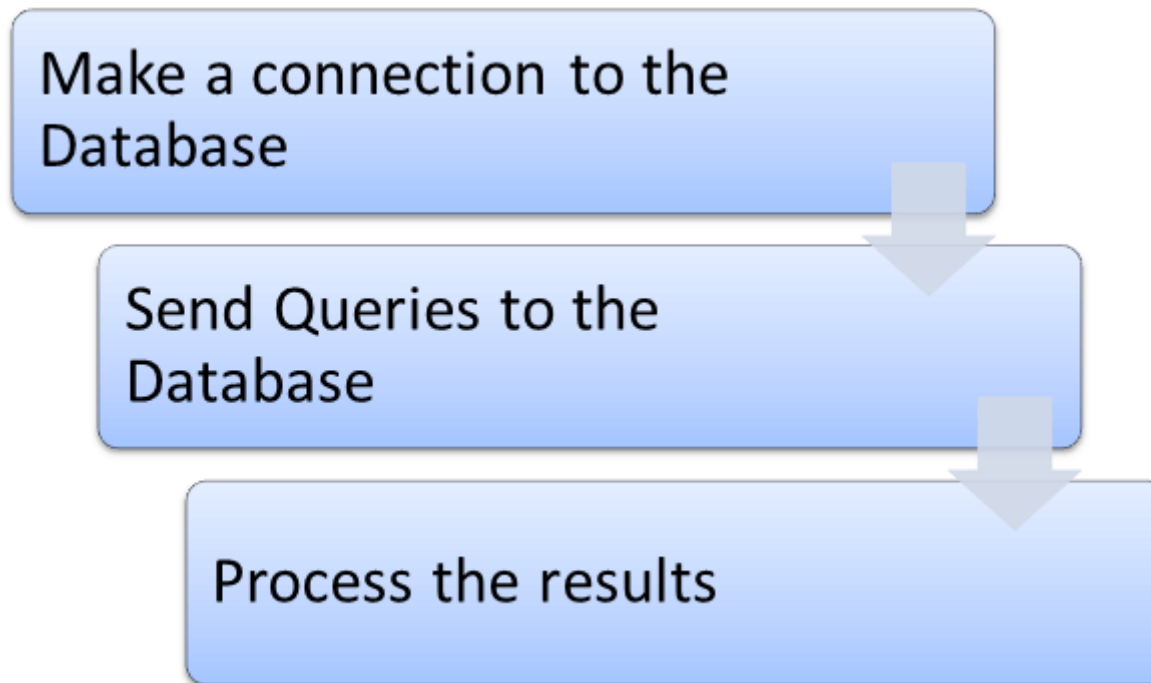
Selenium Webdriver is limited to [Testing](#) your applications using Browser. To use Selenium Webdriver for Database Testing you need to use the JDBC ("Java Database Connectivity").

JDBC (Java Database Connectivity) is a [SQL](#) level API that allows you to execute SQL statements. It is responsible for the connectivity between the [Java](#) 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 order to test your Database using Selenium, you need to observe the following 3 steps1. Make a connection to the Database2. Send Queries to the Database3. Process the results



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 be jdbc: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

Method name	Description
String getString()	Method is used to fetch the string type data from the result set
int getInt()	Method is used to fetch the integer type data from the result set
double getDouble()	Method is used to fetch the double type data from the result set
Date getDate()	Method is used to fetch the Date type object from the result set
boolean next()	Method is used to move to the next record in the result set
boolean previous()	Method is used to move to the previous record in the result set
boolean first()	Method is used to move to the first record in the result set
boolean last()	Method is used to move to the last record in the result set
boolean absolute(int rowNumber)	Method is used to move to the specific record in the result set

Example of Database Testing with Selenium

Step 1) Install [MySQL Server](#) and [MySQL Workbench](#)

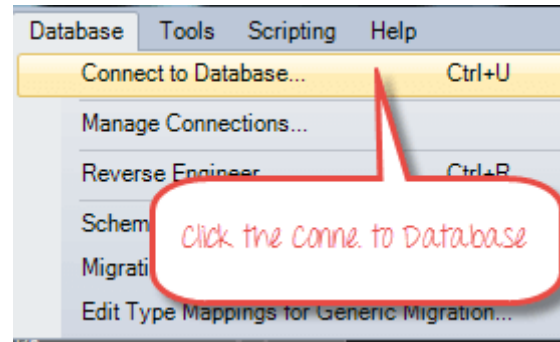
While installing MySQL Server, please note the database

- 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



In the next screen,

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

The screenshot shows the 'Connect to Database' dialog box with the following fields and annotations:

- Stored Connection:** A dropdown menu showing 'Local instance MySQL56' with a red circle '1' next to it.
- Connection Method:** A dropdown menu showing 'Standard (TCP/IP)' with a red circle '2' next to it.
- Parameters tab:** The 'Parameters' tab is selected, with 'SSL' and 'Advanced' tabs also visible.
- Hostname:** A text field containing 'localhost'.
- Port:** A text field containing '3036' with a red circle '2' next to it.
- Username:** A text field containing 'root' with a red circle '3' next to it.
- Password:** A text field with a 'Store in Vault ...' button (annotated with a red circle '4') and a 'Clear' button.
- Default Schema:** An empty text field.
- OK and Cancel buttons:** Located at the bottom right, with a red circle '5' next to the 'OK' button.

Annotations and instructions:

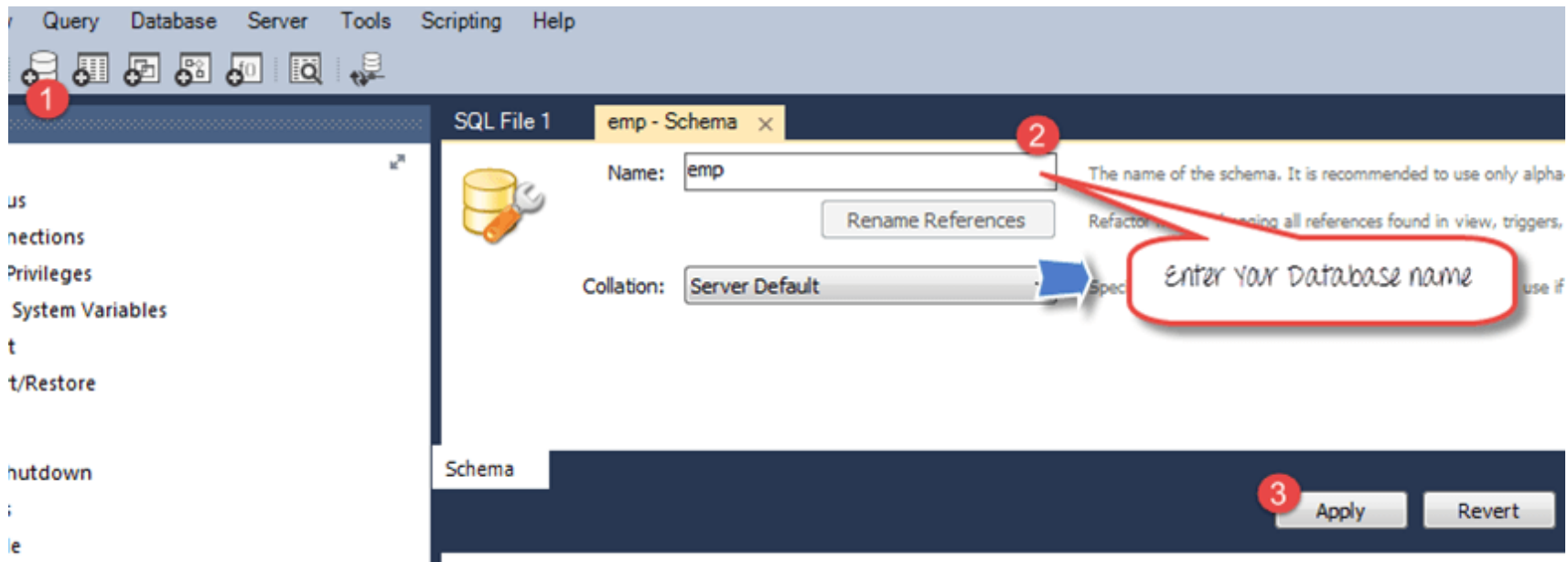
- A red circle '1' is next to the 'Stored Connection' dropdown.
- A red circle '2' is next to the 'Port' field.
- A red circle '3' is next to the 'Username' field.
- A red circle '4' is next to the 'Store in Vault ...' button.
- A red circle '5' is next to the 'OK' button.
- A red circle '2' is next to the 'Port' field.
- A red circle '3' is next to the 'Username' field.
- A red circle '4' is next to the 'Store in Vault ...' button.
- A red circle '5' is next to the 'OK' button.
- A red circle '2' is next to the 'Port' field.
- A red circle '3' is next to the 'Username' field.
- A red circle '4' is next to the 'Store in Vault ...' button.
- A red circle '5' is next to the 'OK' button.

Instructions:

- Enter Your Port Number

Step 3) To Create Database,

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



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

Navigator: MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

SCHEMAS

Filter objects

- emp **1**
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sakila

SQL File 1 emp - Schema employee - Table x

Table Name: employee **2** Schema: emp

Collation: Schema Default InnoDB

Comments:

3

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
Name	VARCHAR(30)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Age	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

4

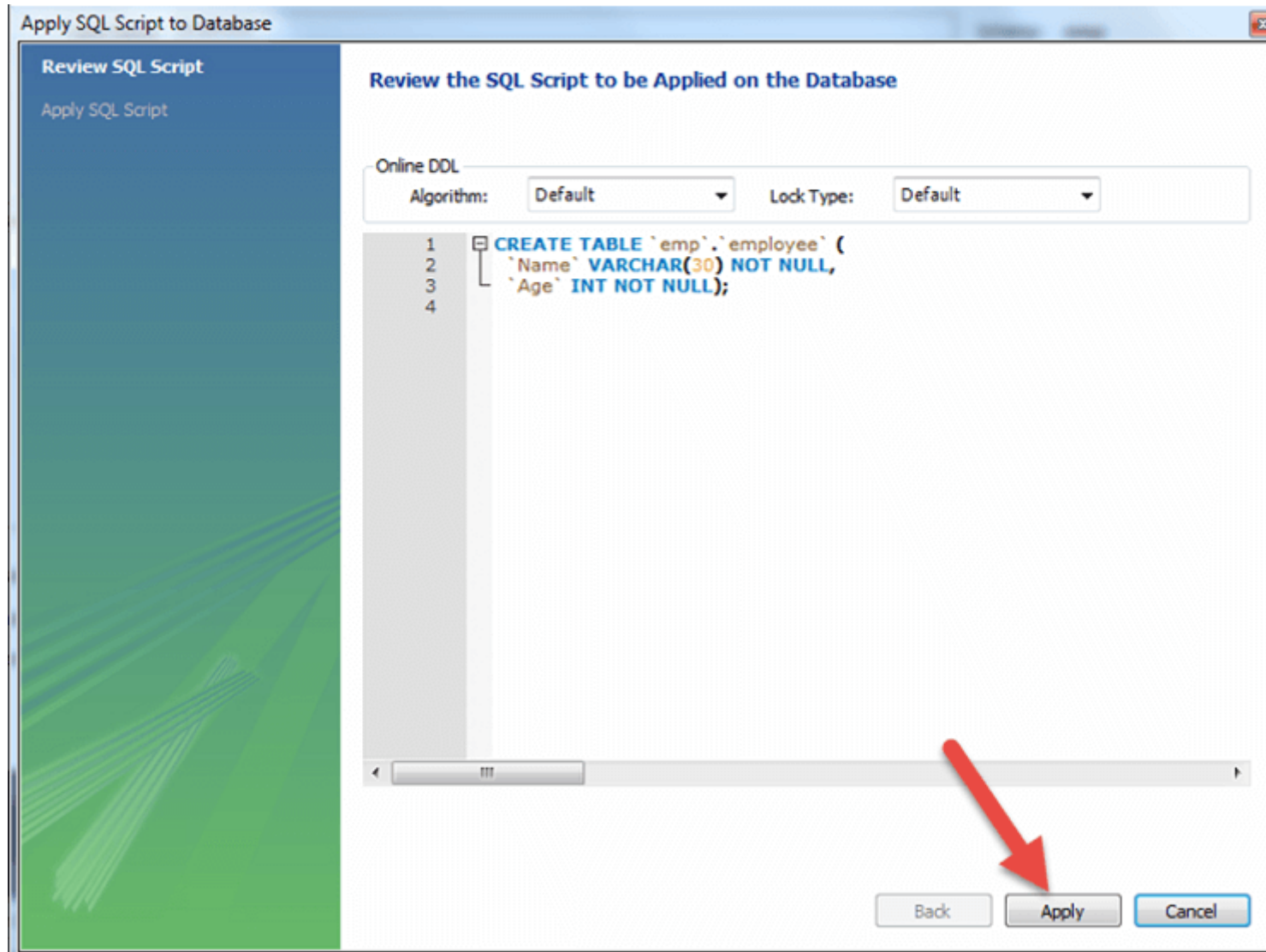
Enter Your Table Name

Enter Your col. Name and Data Type

Columns Indexes Foreign Keys Triggers Partitioning Options

Apply Revert

You will see the following pop-up. Click Apply



Step 5) We will create following data

Name	Age
Top	25

Nick

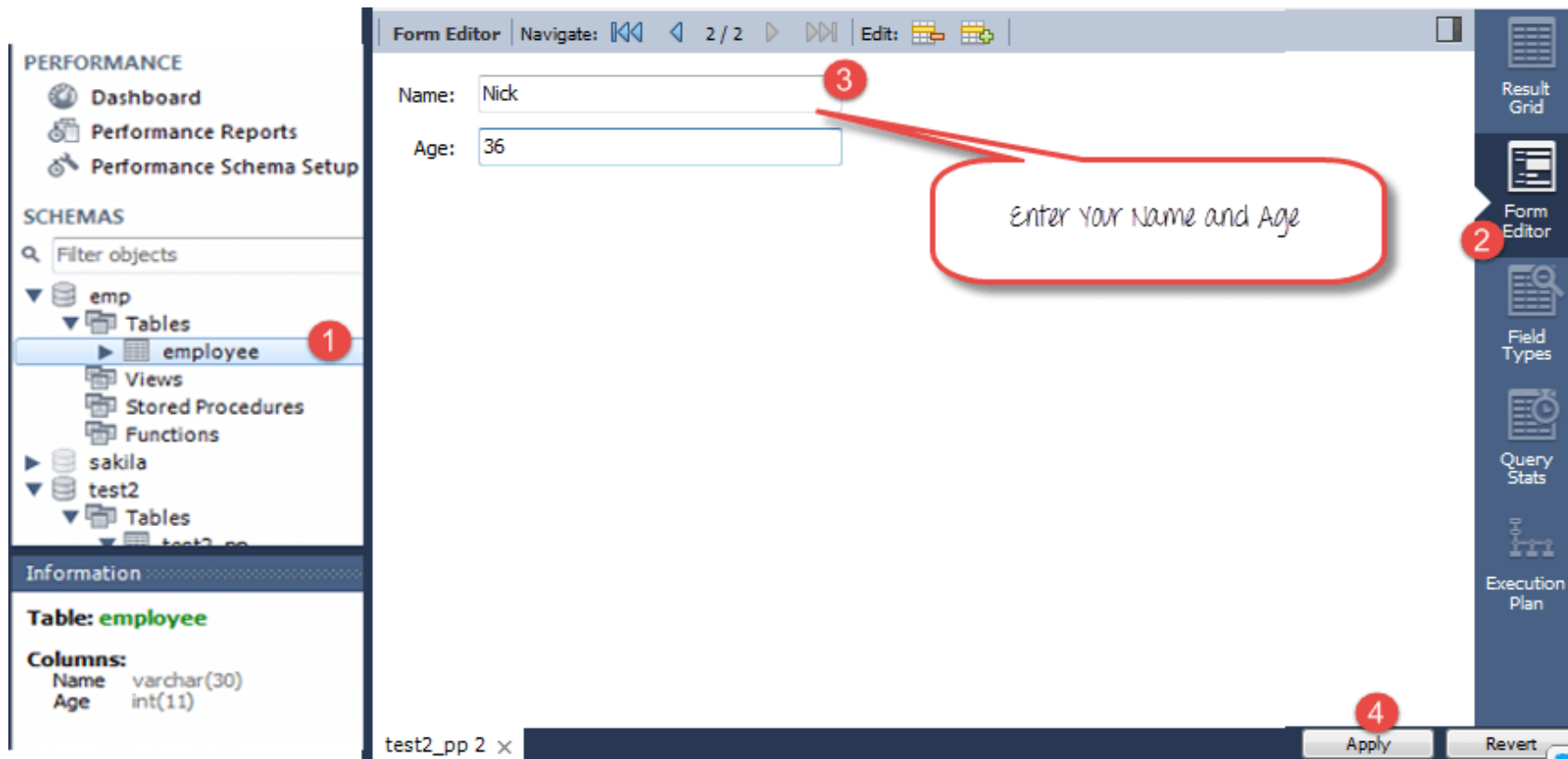
36

Bill

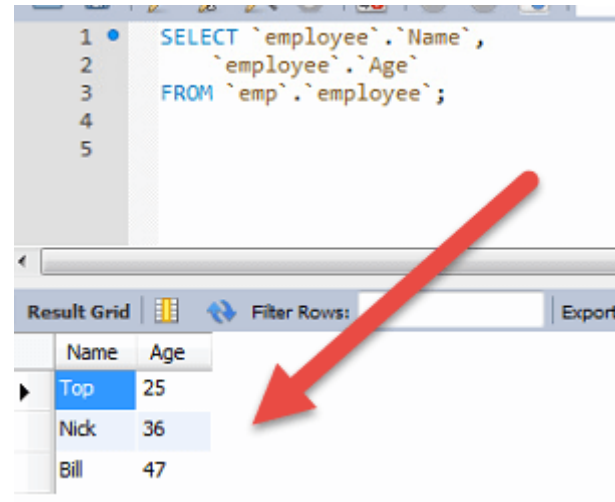
47

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



Repeat the process until all data is created



Step 6) Download the MySQL JDBC connector [here](#)

[Project Home](#) **Downloads** [Wiki](#) [Issues](#) [Source](#) [Export to GitHub](#)

Search Current download: for

Download: **the driver for connecting MySQL to Java**
7 people starred this download

led by: [saloni09...@iiitd.ac.in](#)
ed: Oct 27, 2011
led: Oct 27, 2011
oads: 108153

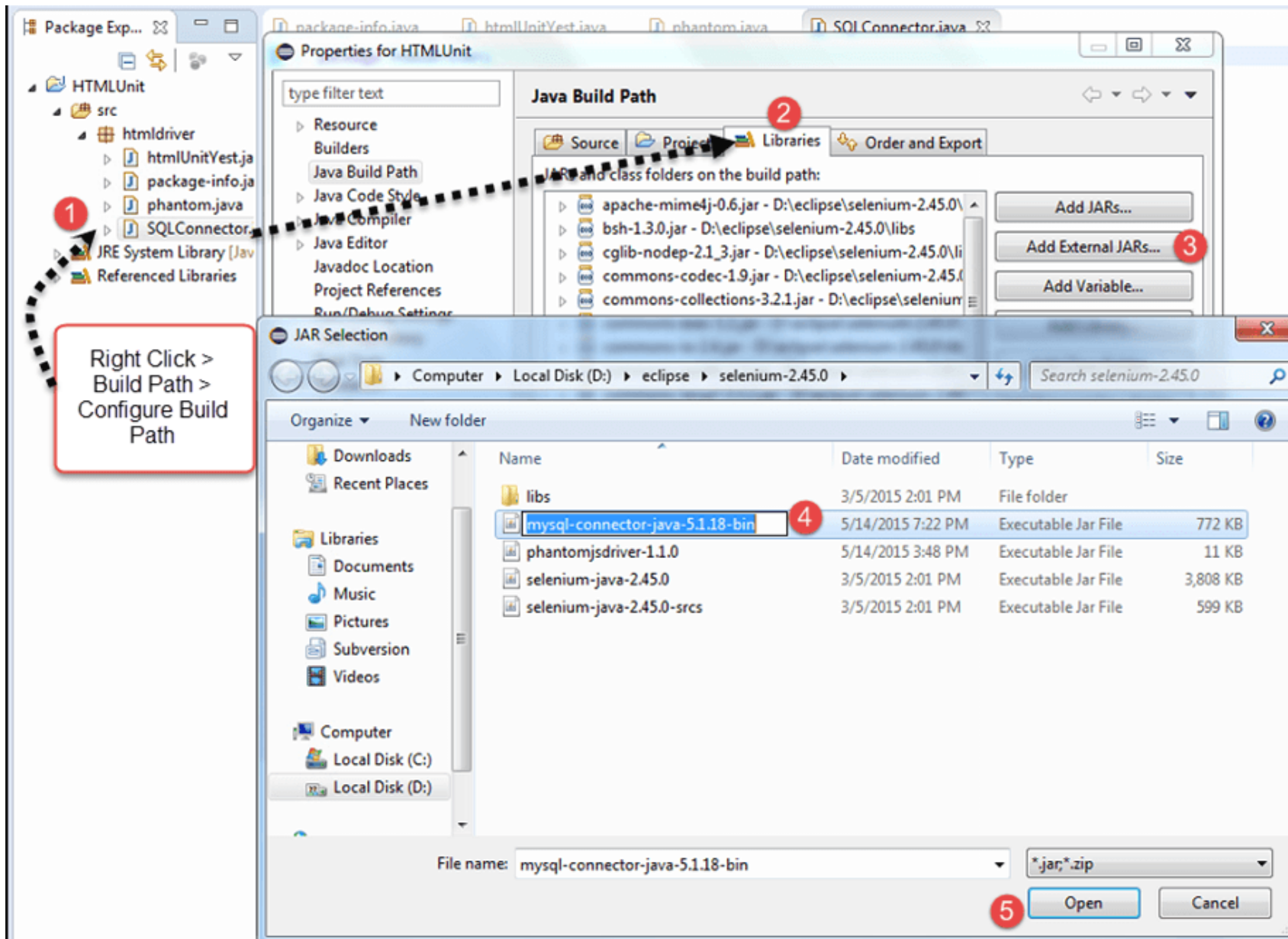
File:  [mysql-connector-java-5.1.18-bin.jar](#) 771 KB

Description: This is a jar file which needs to be copied to the classpath. These are the steps to follow.
Right-click on the project name in eclipse.
Click on Build Path -> Configure Build Path.
Choose Library
Add External Jar
Browse and say Open
Open
You're Done.
Your program can now access the Jar file.

SHA1 Checksum: 85dfedad243dc0303ad7ae3a323c39421d220690 [What's this?](#)

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



Step 8) Copy the following code into the editor

```
Package  htmdldriver;
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

```
1 package htmldriver;
2 import java.sql.Connection;
3 import java.sql.Statement;
4 import java.sql.ResultSet;
5 import java.sql.DriverManager;
6 import java.sql.SQLException;
7
8 public class SQLConnector {
9
10     public static void main(String[] args) throws ClassNotFoundException, SQLException {
11
12         //Connection URL Syntax: "jdbc:mysql://ipaddress:portnumber/db_name"
13         String dbUrl = "jdbc:mysql://localhost:3036/emp";
14
15         //Database Username
16         String username = "root";
17
18         //Database Password
19         String password = "guru99";
20
21         //Query to Execute
22         String query = "select * from employee;";
23
24         //Load mysql jdbc driver
25         Class.forName("com.mysql.jdbc.Driver");
26
27         //Create Connection to DB
28         Connection con = DriverManager.getConnection(dbUrl,username,password);
29
30         //Create Statement Object
```

Problems @ Javadoc Declaration Console

<terminated> SQLConnector [Java Application] C:\Program Files\Java\jre1.8.0_45\bin\javaw.exe (May 15, 2015, 3:19:36 PM)

Top	25
Nick	36
Bill	47

Summary

- In order to test Database using Selenium you need to
 1. Make a connection to the Database
 2. Send Queries to the Database
 3. Process the results

- The Syntax to connect to Database is
 - `DriverManager.getConnection(URL, "userid", "password")`
- You will also need the Statement Object to send queries
 - `Statement stmt = con.createStatement();`
- To send the query to database use 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

[< Prev](#)

[Next >](#)

RELATED ARTICLES

JAVA TUTORIALS



Creating Your First Java Program

AWS



Creating a Spot Amazon EC2 instance

LINUX



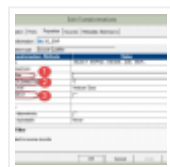
Introduction to PERL Programming

SAP - CRM



How SAP CRM Interaction Center works

INFORMATICA



Aggregator Transformation in Informatica

SAP - HR



Bank Transfer using SAP DME File