**JDBC –SQL( JAVA INTEGRITY)**

**Connecting to MySQL Using JDBC Driver**

**Main Steps:**

1. Load the MySQL Connector/J into your program.

2. Create a new Connection object from the DriverManager class. Then you can use this Connection object to execute queries.

**Step-I:**

## Loading MySQL Connector/J into your program

Add JAR folder button, browse to the location where you installed MySQL Connector/J, and choose the JAR file.

**Step-II:**

## Connecting to MySQL database

First, you need to import three classes: SQLException, DriverManager, and Connection from the java.sql.\* package.

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

Code language: Java (java)

Second, you call the getConnection() method of the DriverManager class to get the Connection object. There are three parameters you need to pass to the getConnection() method:

1. url: the database URL in the form jdbc:subprotocol:subname. For MySQL, you use the jdbc:mysql://localhost:3306/mysqljdbc i.e., you are connecting to the MySQL with server name localhost, port 3006, and database mysqljdbc.
2. user: the database user that will be used to connect to MySQL.
3. password: the password of the database user.

When connecting to MySQL, anything could happens e.g., database server is not available, wrong user name or password, etc. in such cases, JDBC throws a SQLException . Therefore, when you create a Connection object, you should always put it inside a try catch block. Also you should always close the database connection once you complete interacting with database by calling close() method of the Connection object.

t is automatically calls the close() method of the Connection object once program finishes. As you can see it’s cleaner and more elegant. However…

It is not secure as well as flexible when you hard coded the database parameters inside the code like above. In case you change the database server or password; you have to change the code, compile it again, which is not a good design.

To avoid hard coding all the database parameters in the code, you can use a Java properties file to store them. In case of changes, you just need to change them in the properties file and you don’t have to recompile the code.

For each interaction with MySQL database, you need to create a new connection. You would have the same piece of code for doing this in all places. Rather than doing this, you can create a new class for handing connection creation:

**Packages Required to connect Mysql with Java:**

**package** org.mysqltutorial;

**import** java.io.FileInputStream;

**import** java.io.IOException;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**import** java.util.Properties;

**package** org.mysqltutorial;

**import** java.sql.Connection;

**import** java.sql.SQLException;