STEP 1:

//Start hive server (command) on terminal:

\$hive -service hiveserver2 &

//type ps to check if server is running(we will get java pid).

\$ps

STEP 2:

1. Create a Maven Project.

```
Right click on Project Explorer -> New-> Others-> Maven Project-> Next.

Check on "Create a Simple Project (Skip Archetype Selection)"

Group Id: com.hlc.java2hive

Artifact Id: JavaToHiveConnection
```

2. Add the below configuration in your pom.xml under ct>...

3. Add a Java class.

Expand Your Project and create a Java class under src/main/java
Package Name : com.hlc.hivejava
Class Name : 1HiveConnectionTest

4. Add main() to your class and write the below logic

Note: hduser is the user using which you would be connection to <u>hive.when</u> using CLI also we used hduser as the current user only and all the tables are created using that.

REFERENCE :KAUSHIK SHAKKARI EMAIL : kaushik.shakkari@gmail.com

hivejdbc.java

```
package com.hlc.java2hive;
import java.sql.SQLException;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.DriverManager;
public class hiveJdbc {
       private static String driverName = "org.apache.hive.jdbc.HiveDriver";
         * @param args
         * @throws SQLException
         public static void main(String[] args) throws SQLException {
           try {
           Class.forName(driverName);
          } catch (ClassNotFoundException e) {
           // TODO Auto-generated catch block
           e.printStackTrace();
           System.exit(1);
          }
           int col = 1;
           //String str = blis_chargeactivation.bill();
           String str = xxrpt_art.bill1();
           System.out.println(str);
         //replace "hive" here with the name of the user the gueries should run as
          Connection con = DriverManager.getConnection("jdbc:hive2://localhost:10000/default",
"bd", "");
          Statement stmt = con.createStatement();
         // show tables
         // String sql = "show tables "" + tableName + """;
          String sql = str;
          ResultSet res = stmt.executeQuery(sql);
          while(res.next()){
            System.out.println(res.getString(col));
           }
          res.close();
         stmt.close();
         con.close();
          System.out.println("DONE");
```

REFERENCE: KAUSHIK SHAKKARI EMAIL: kaushik.shakkari@gmail.com

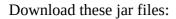
```
}
```

xxptr_art.java

```
package com.hlc.java2hive;
import java.util.Scanner;
public class xxrpt art
{
      public static String bill1()
            String str = null;
            Scanner <u>sc</u> = new Scanner(System.in);
            int choice;
            System.out.println("enter the choice\n1.create table\n2.load
data\n3.print all data\n4.print a column\n5.show all tables");
            choice = sc.nextInt();
            if (choice == 1)
                  str = "CREATE TABLE artemisites(HGSSITEID INT, CREATEDATE
STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ' , '";
            else if(choice == 2)
                  str = "load data local inpath '/home/bd/Downloads/IECMeeting-
Sampledata/XXRPTH/xxrpt artemissites.csv' into table artemisites";
            else if(choice == 3)
                  str = "select * from artemisites";
            else if(choice == 4)
                  str = "select HGSSITEID from artemisites";
            else if(choice == 5)
                  str = "show tables";
            return str;
}
```

STEP 3:

REFERENCE :KAUSHIK SHAKKARI EMAIL : kaushik.shakkari@gmail.com



- (1).hadoop-common-2.7.3.jar
- (2).hive-exec-2.1.1-standalone.jar
- (3).hive-jdbc-2.1.1-standalone.jar

Add these external jars in eclipse

goto build path then configure build path then click on libraries and add these external jars manually.

STEP 4:

Warning:

Delete multiple binding files following warning path.

REFERENCE :KAUSHIK SHAKKARI EMAIL : kaushik.shakkari@gmail.com