

JAVA WITH JDBC

EXERCISE

CONNECTION POOL WITH JDBC AND APACHE

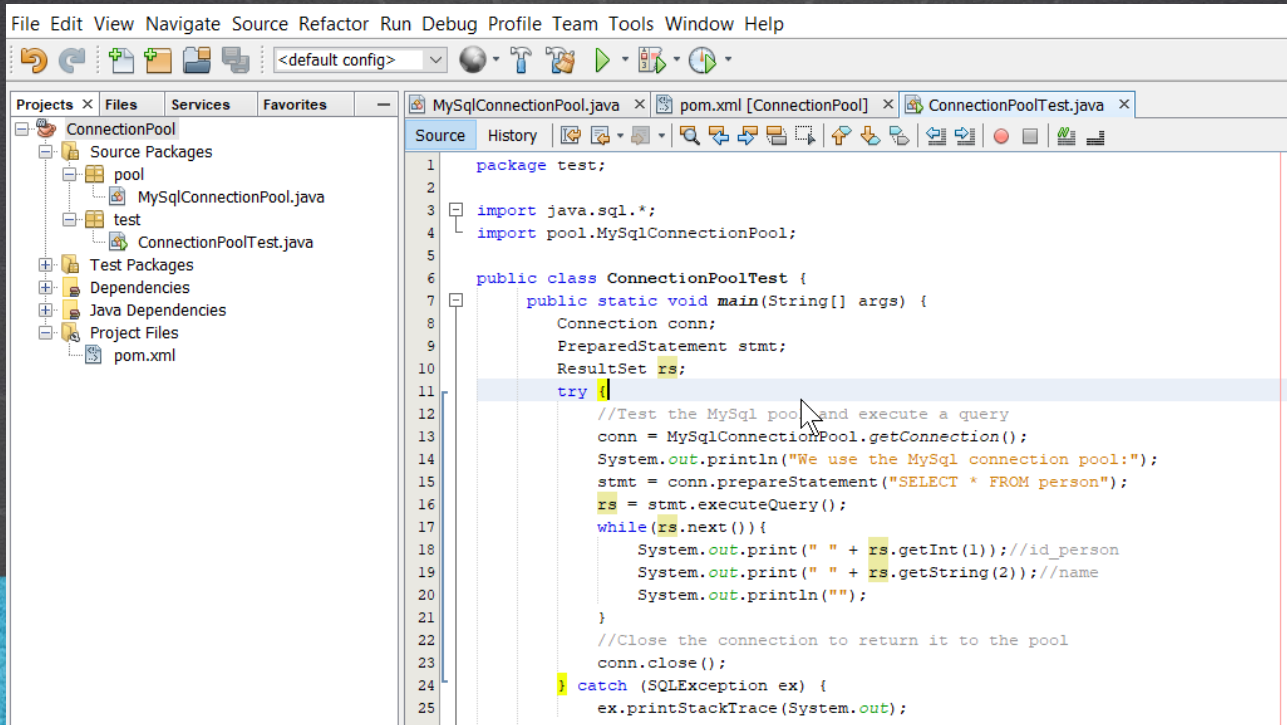


JAVA WITH JDBC

www.globalmentoring.com.mx

EXERCISE OBJECTIVE

Create the connection pool exercise. At the end we should observe the following:



The screenshot shows an IDE window with the following structure:

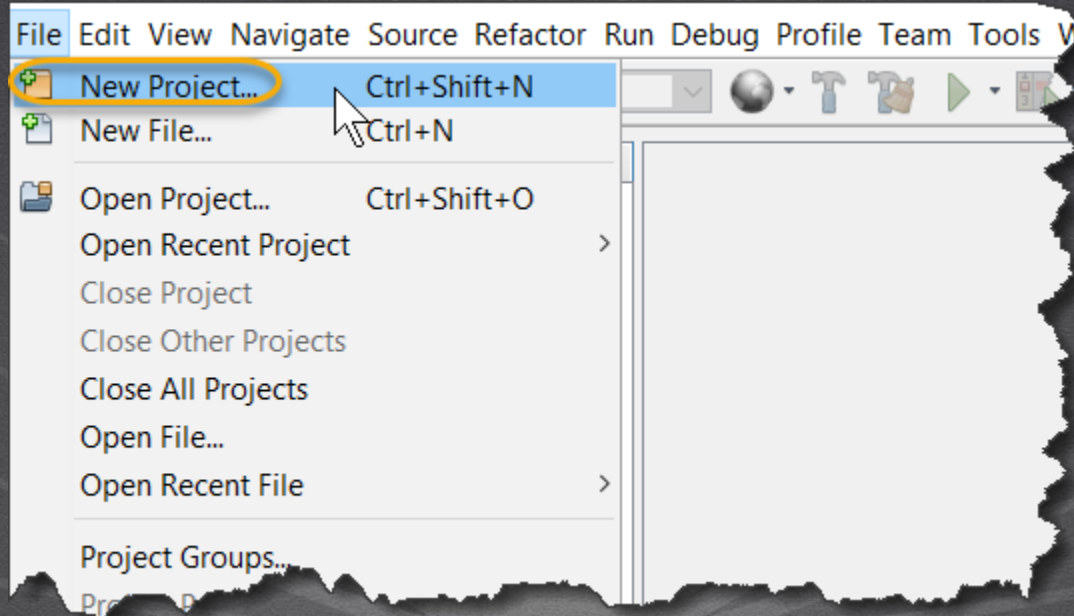
- Projects:** ConnectionPool
 - Source Packages
 - pool
 - MySQLConnectionPool.java
 - test
 - ConnectionPoolTest.java
 - Test Packages
 - Dependencies
 - Java Dependencies
 - Project Files
 - pom.xml

The **Source** tab is active, showing the code for `ConnectionPoolTest.java`:

```
1 package test;
2
3 import java.sql.*;
4 import pool.MySqlConnectionPool;
5
6 public class ConnectionPoolTest {
7     public static void main(String[] args) {
8         Connection conn;
9         PreparedStatement stmt;
10        ResultSet rs;
11        try {
12            //Test the MySql pool and execute a query
13            conn = MySqlConnectionPool.getConnection();
14            System.out.println("We use the MySql connection pool:");
15            stmt = conn.prepareStatement("SELECT * FROM person");
16            rs = stmt.executeQuery();
17            while(rs.next()){
18                System.out.print(" " + rs.getInt(1)); //id_person
19                System.out.print(" " + rs.getString(2)); //name
20                System.out.println("");
21            }
22            //Close the connection to return it to the pool
23            conn.close();
24        } catch (SQLException ex) {
25            ex.printStackTrace(System.out);
26        }
27    }
28 }
```


1. CREATE A NEW PROJECT

Create a new project:

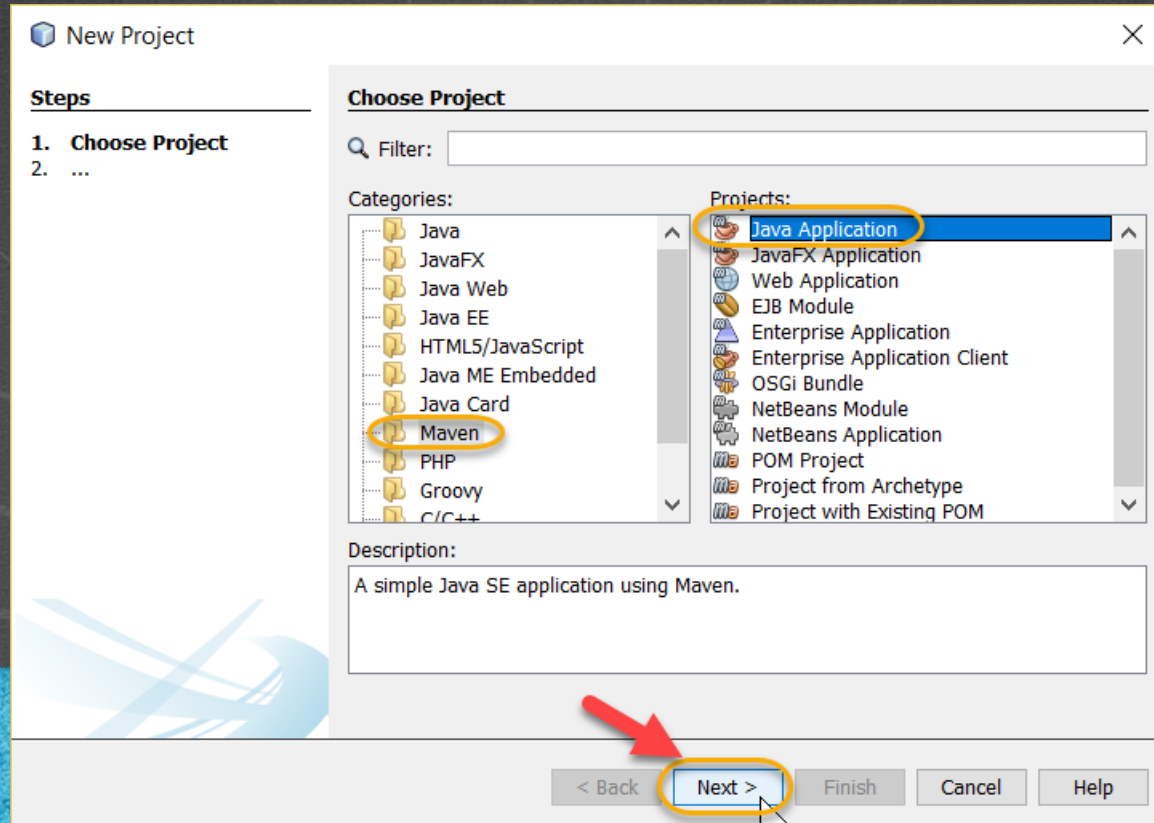


JAVA WITH JDBC

www.globalmentoring.com.mx

1. CREATE A NEW PROJECT

Create a new project:



1. CREATE A NEW PROJECT

Create a new project:

New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name:

Project Location:

Project Folder:

Artifact Id:

Group Id:

Version:

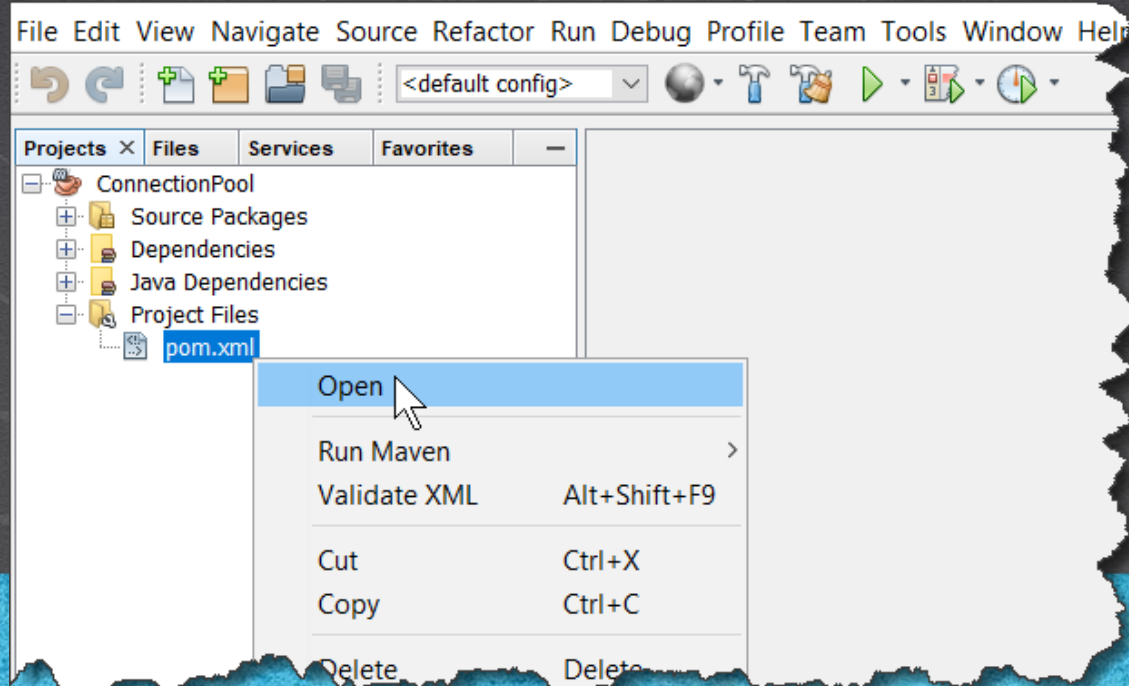
Package: (Optional)

JAVA WITH JDBC

www.globalmentoring.com.mx

2. MODIFY THE POM.XML

Modify the pom.xml to add the mysql.jar, and other dependencies for creating the connection pool with apache libraries:



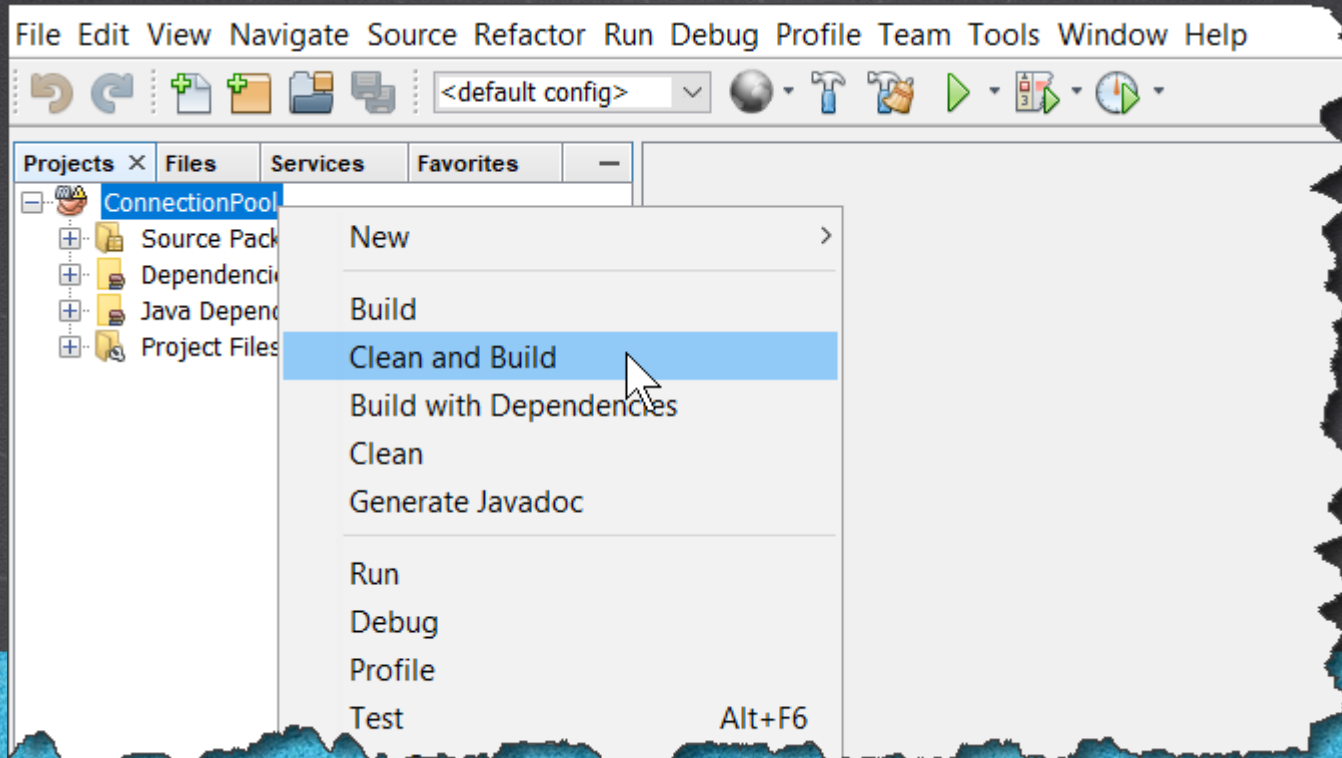
2. MODIFY THE CODE

pom.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>mx.com.gm</groupId>
    <artifactId>ConnectionPool</artifactId>
    <version>1</version>
    <packaging>jar</packaging>
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
    </properties>
    <dependencies>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>5.1.46</version>
        </dependency>
        <dependency>
            <groupId>org.apache.commons</groupId>
            <artifactId>commons-dbcp2</artifactId>
            <version>2.5.0</version>
        </dependency>
    </dependencies>
</project>
```

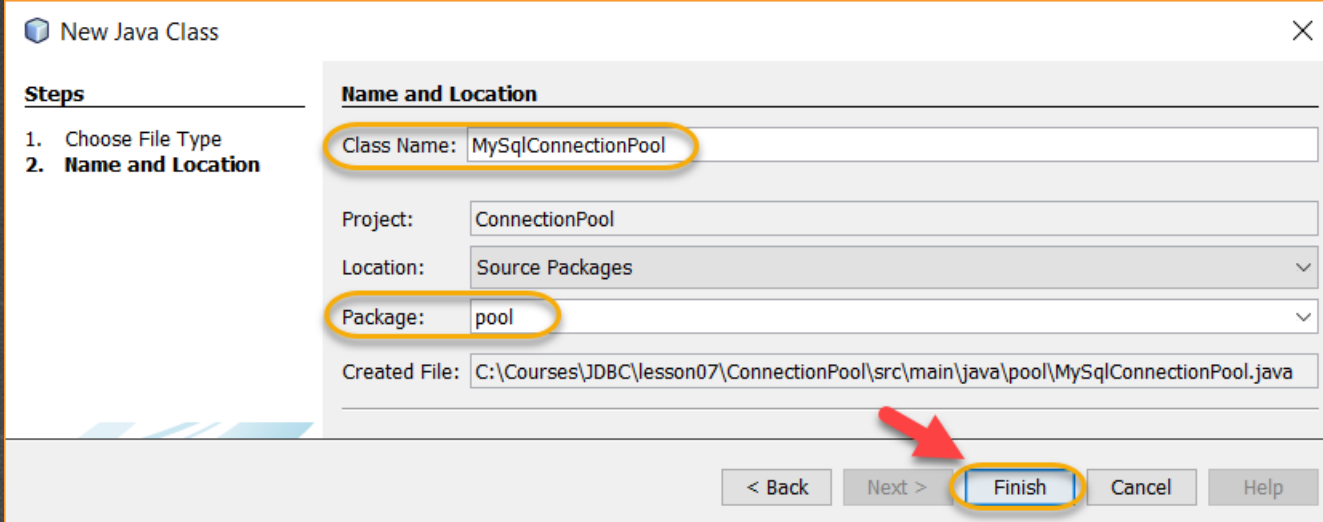
3. CLEAN & BUILD

Execute the clean & build option:



4. CREATE A NEW CLASS

Create a new class:



New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

< Back Next > **Finish** Cancel Help

JAVA WITH JDBC

www.globalmentoring.com.mx

5. MODIFY THE CODE

MysqlConnectionPool.java:

```
package pool;

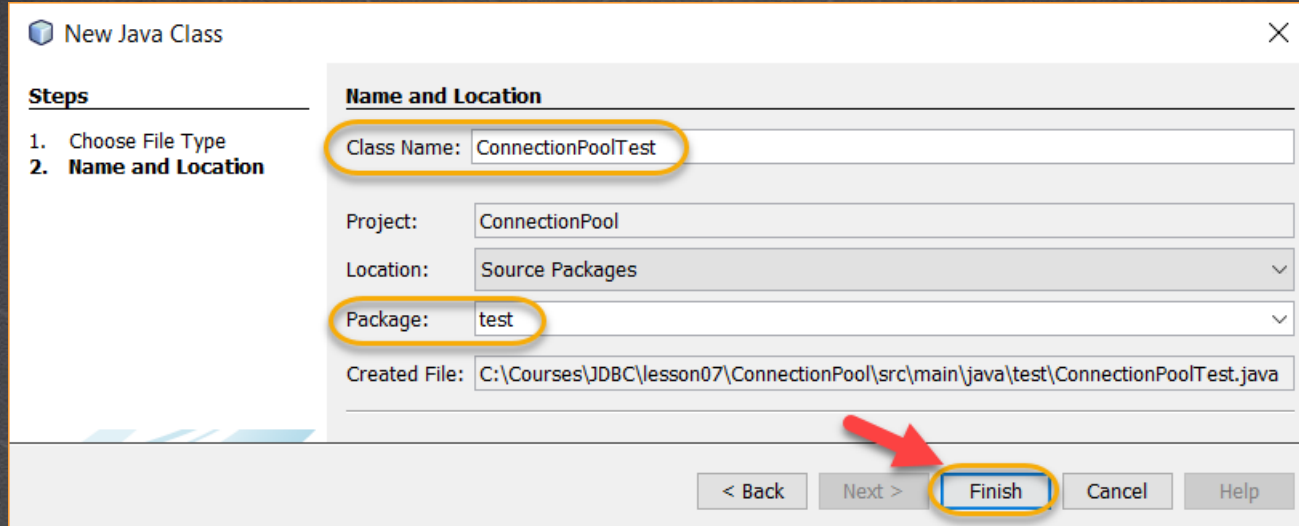
import java.sql.*;
import javax.sql.DataSource;
import org.apache.commons.dbcp2.BasicDataSource;

public class MySqlConnectionPool {
    public static DataSource getDataSource() {
        BasicDataSource ds = new BasicDataSource();
        ds.setDriverClassName("com.mysql.jdbc.Driver");
        ds.setUsername("root");
        ds.setPassword("admin");
        ds.setUrl("jdbc:mysql://localhost:3306/test?useSSL=false");
        //We define the size of the connection pool
        ds.setInitialSize(5); //5 Initial connections
        return ds;
    }

    public static Connection getConnection() throws SQLException{
        return getDataSource().getConnection();
    }
}
```

6. CREATE A NEW CLASS

Create a new class:



New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

< Back Next > **Finish** Cancel Help

JAVA WITH JDBC

www.globalmentoring.com.mx

7. MODIFY THE CODE

ConnectionPoolTest.java:

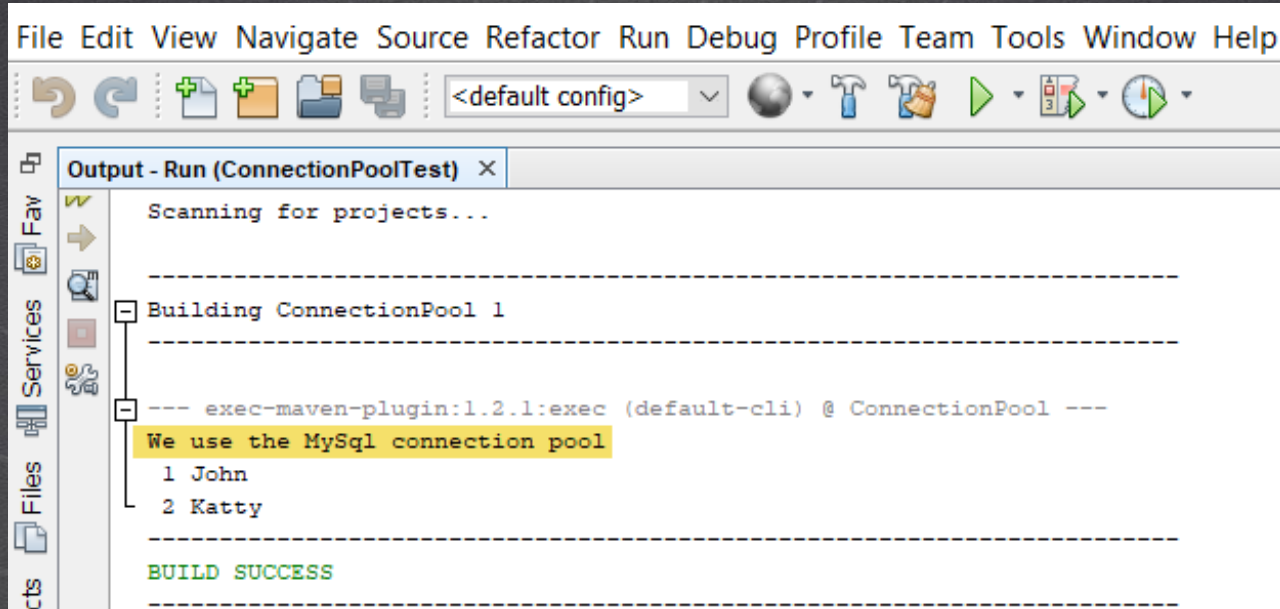
```
package test;

import java.sql.*;
import pool.MySqlConnectionPool;

public class ConnectionPoolTest {
    public static void main(String[] args) {
        Connection conn;
        PreparedStatement stmt;
        ResultSet rs;
        try {
            //Test the MySQL pool and execute a query
            conn = MySqlConnectionPool.getConnection();
            System.out.println("We use the MySQL connection pool:");
            stmt = conn.prepareStatement("SELECT * FROM person");
            rs = stmt.executeQuery();
            while(rs.next()){
                System.out.print(" " + rs.getInt(1)); //id_person
                System.out.print(" " + rs.getString(2)); //name
                System.out.println("");
            }
            //Close the connection to return it to the pool
            conn.close();
        } catch (SQLException ex) {
            ex.printStackTrace(System.out);
        }
    }
}
```

8. EXECUTE THE PROJECT

The result is as follows:



The screenshot shows the 'Output - Run (ConnectionPoolTest)' window in an IDE. The output text is as follows:

```
Scanning for projects...

-----
Building ConnectionPool 1
-----
--- exec-maven-plugin:1.2.1:exec (default-cli) @ ConnectionPool ---
We use the MySql connection pool
1 John
2 Katty
-----

BUILD SUCCESS
```

JAVA WITH JDBC

www.globalmentoring.com.mx

EXERCISE CONCLUSION

- With this exercise we have put into practice the concept of Connection Pool, with which we can reuse and streamline the use of available connections for our Java applications.
- We will do the same with the support of application servers such as Glassfish or Jboss in later courses.



JAVA WITH JDBC

www.globalmentoring.com.mx

ONLINE COURSE

JAVA WITH JDBC

By: Eng. Ubaldo Acosta



JAVA WITH JDBC

www.globalmentoring.com.mx