

You have **Eclipse Temurin JDK 25 LTS**, so Java itself isn't the problem. ☐

The remaining issue is simply **Maven**.

1. Check whether Maven is already extracted

Run this in PowerShell:

```
Test-Path "C:\Program Files\Apache\maven\bin\mvn.cmd"
```

If it says:

True

Maven is installed, and we just need to fix your PATH.

If it says:

False

Maven isn't in that location yet.

2. If you haven't downloaded Maven yet

Download the **Binary zip archive** from:

[Apache Maven Downloads](#)

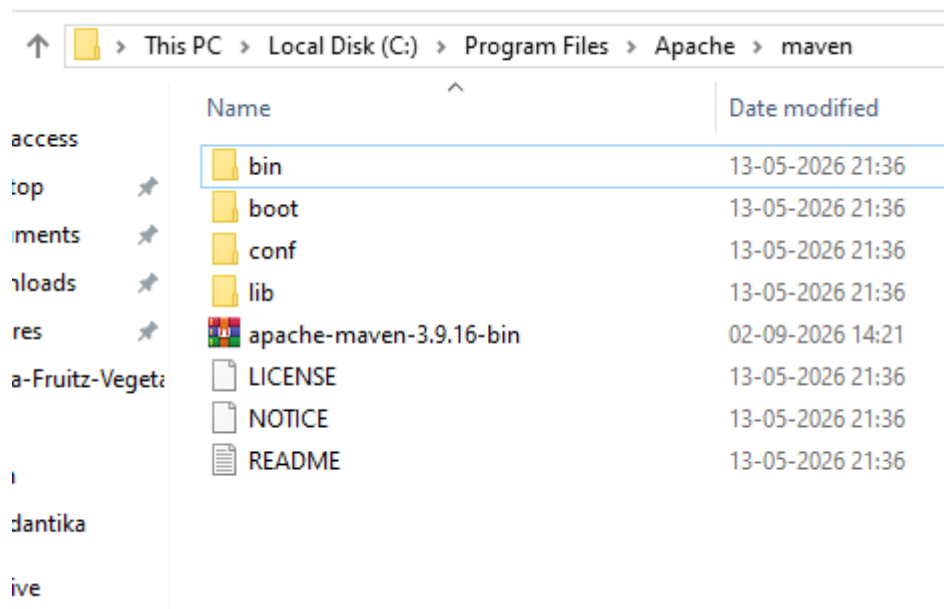
Extract it somewhere simple, for example:

```
C:\Program Files\Apache\maven
```

Then make sure this file exists:

```
C:\Program Files\Apache\maven\bin\mvn.cmd
```

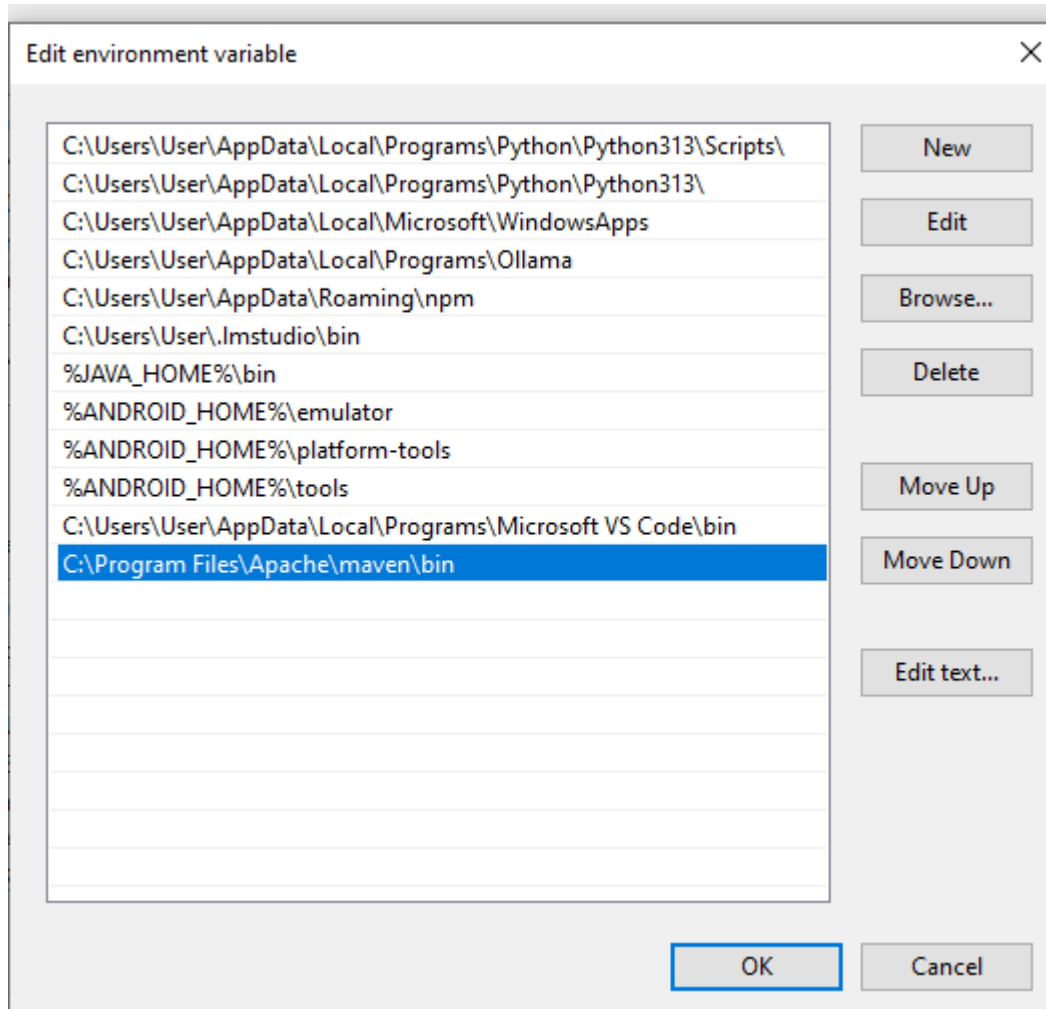
As shown below



3. Add it to PATH

Once that file exists, add this to your Windows **Path**:

C:\Program Files\Apache\maven\bin



Then **completely close and reopen VS Code**.

Run:-

```
mvn -version
```

You should get something like:

```
Apache Maven 3.9.x
Maven home: C:\Program Files\Apache\maven
Java version: 25.0.4.1
```

Next: Create your Java + MySQL project

Since your goal is to connect Java to MySQL, I'd recommend doing this next:

1. Create a Maven project

In VS Code:

Ctrl + Shift + P

Search:

Maven: New Project

Choose:

maven-archetype-quickstart

Then create something like:

Group Id: com.example

Artifact Id: java-demo

And in your project folder in pom.xml these two things important :-

```
<properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>21</maven.compiler.source>
    <maven.compiler.target>21</maven.compiler.target>
</properties>

<dependencies>
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.11</version>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>com.mysql</groupId>
        <artifactId>mysql-connector-j</artifactId>
        <version>9.4.0</version>
    </dependency>
</dependencies>
```

Your Maven project is ready.

Now let's connect **Java** → **MySQL**.

Step 1: Open pom.xml

In VS Code, open:

```
mysql-demo
└─ pom.xml
```

You'll see something similar to:

```
<project>
  ...
  <dependencies>
    ...
  </dependencies>
</project>
```

Inside <dependencies>, add the MySQL JDBC driver:

```
<dependency>
  <groupId>com.mysql</groupId>
  <artifactId>mysql-connector-j</artifactId>
  <version>9.4.0</version>
</dependency>
```

So your dependencies should look something like:

```
<dependencies>

  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>3.8.1</version>
    <scope>test</scope>
  </dependency>

  <dependency>
    <groupId>com.mysql</groupId>
    <artifactId>mysql-connector-j</artifactId>
    <version>9.4.0</version>
  </dependency>

</dependencies>
```

Step 2: Save pom.xml

Press:

Ctrl + S

Maven should automatically download the MySQL driver.

You can also run:

```
mvn clean compile
```

Inside `App.java` file code:-

```
package com.example;

import java.sql.Connection;
import java.sql.DriverManager;

public class App {

    public static void main(String[] args) {

        String url = "jdbc:mysql://localhost:3306/mysql_demo";
        String username = "root";
        String password = "";

        try {
            Connection con = DriverManager.getConnection(
                url,
                username,
                password
            );

            System.out.println("MySQL Connected Successfully!");

            con.close();

        } catch (Exception e) {
            System.out.println("Connection Failed!");
            e.printStackTrace();
        }

    }
}
```

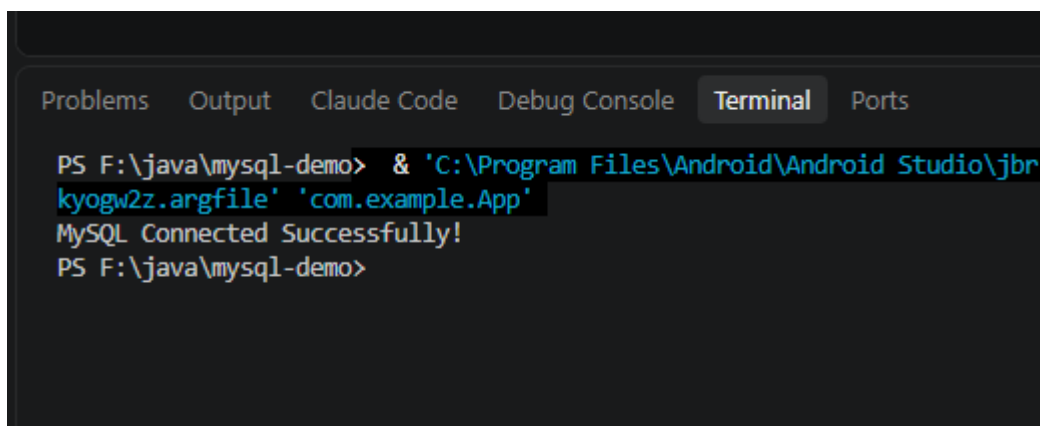
And finally

4. Run

In VS Code terminal:-

```
mvn compile
```

Then click **Run ▶** above main().

A screenshot of a VS Code terminal window. The terminal has tabs for 'Problems', 'Output', 'Claude Code', 'Debug Console', 'Terminal' (which is active), and 'Ports'. The terminal text shows a PowerShell prompt 'PS F:\java\mysql-demo>' followed by a command to run a Java application with a MySQL connection URL. The output shows 'MySQL Connected Successfully!' and the prompt returns to 'PS F:\java\mysql-demo>'.

```
PS F:\java\mysql-demo> & 'C:\Program Files\Android\Android Studio\jbr
kyogw2z.argfile' 'com.example.App'
MySQL Connected Successfully!
PS F:\java\mysql-demo>
```

Note :-

That's it. You **don't need** `Class.forName()` with modern MySQL Connector/J; the JDBC driver is automatically loaded.