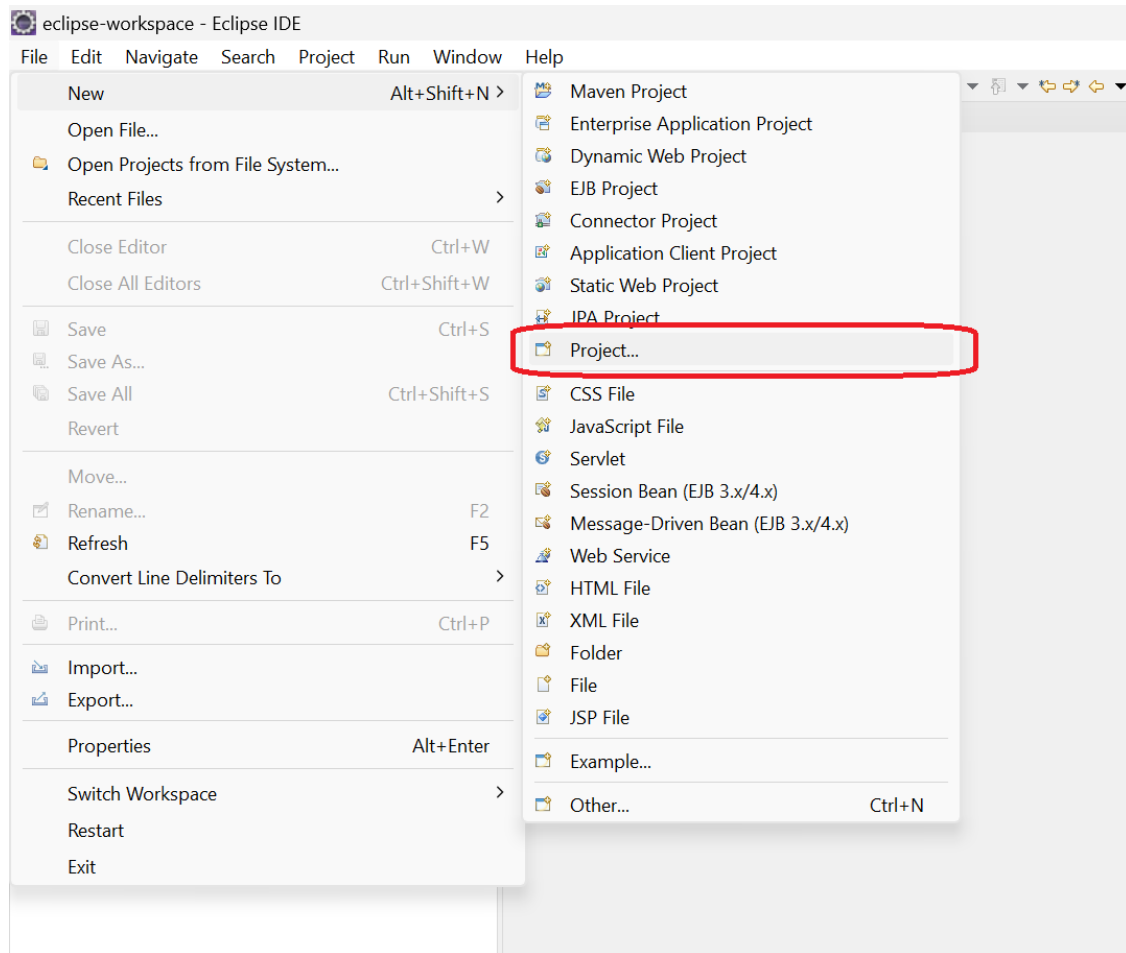
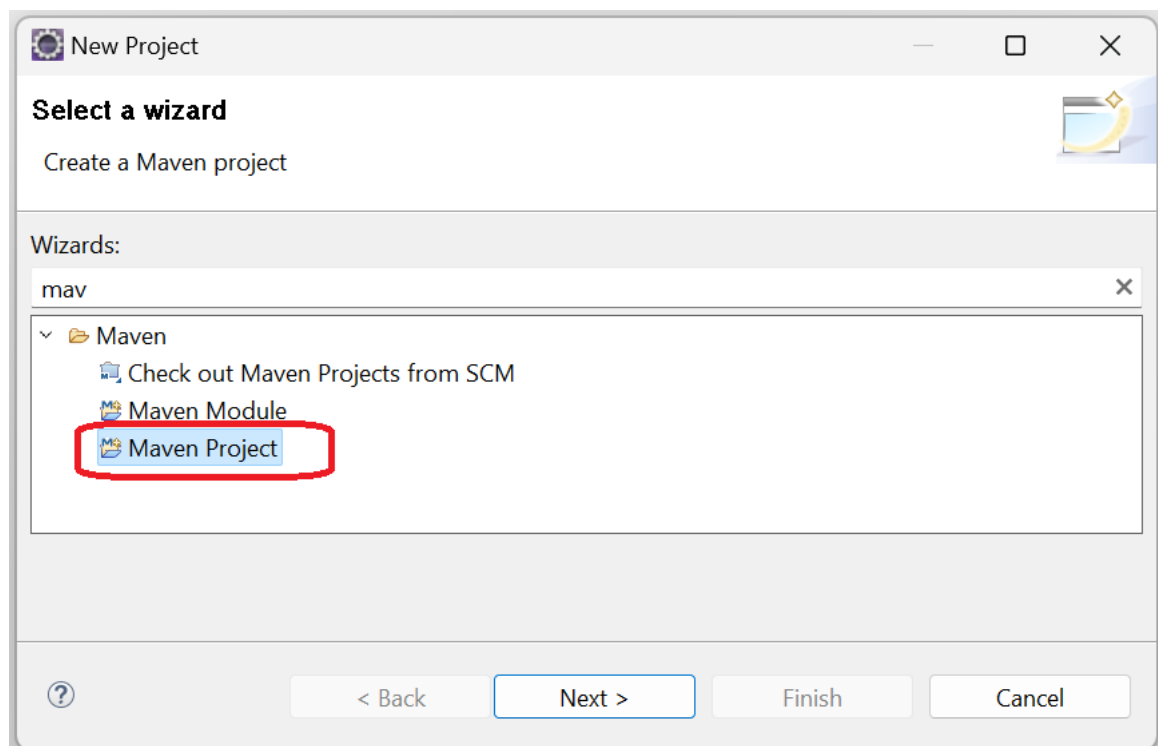


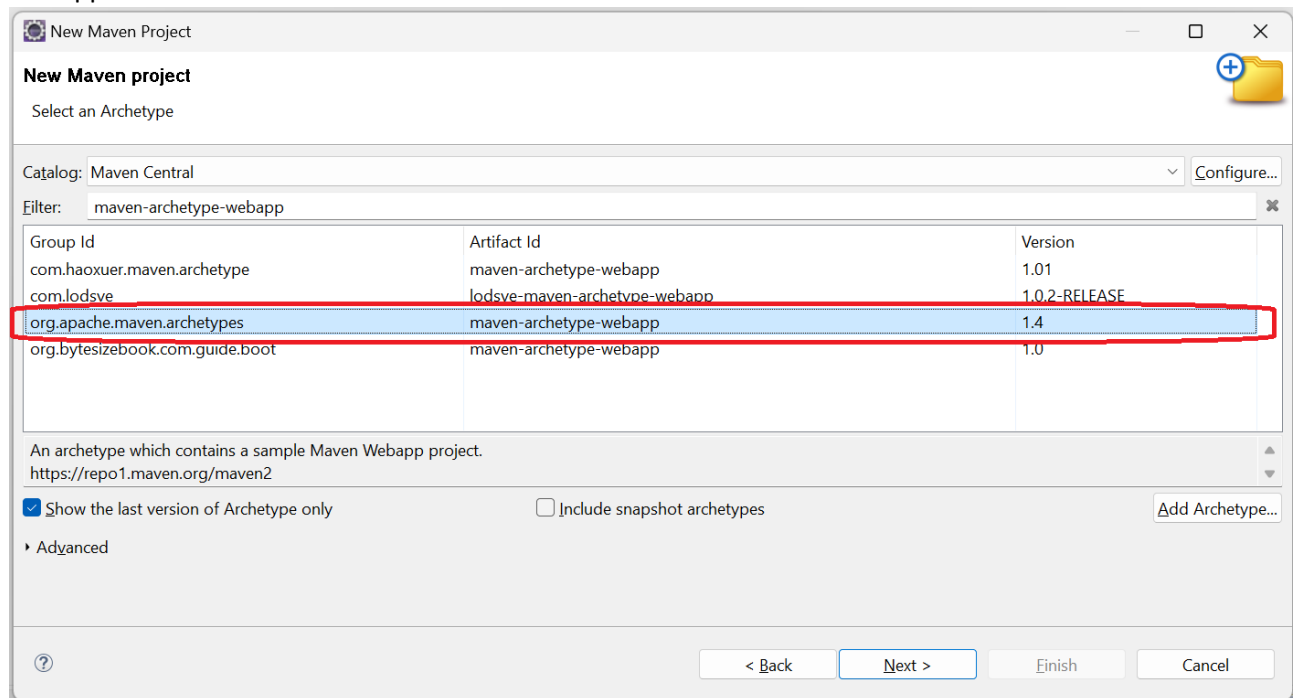
1. Creare un nuovo progetto con Eclipse JEE



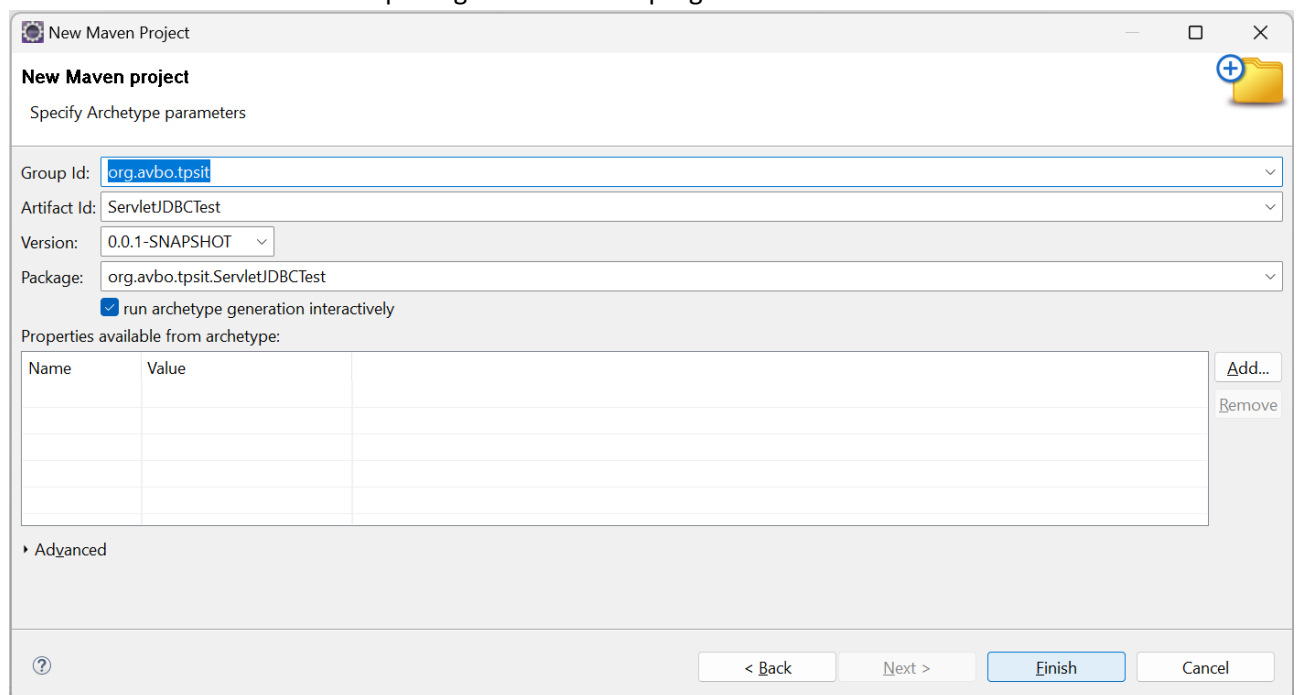
2. Selezionare il template per creare un nuovo progetto Maven



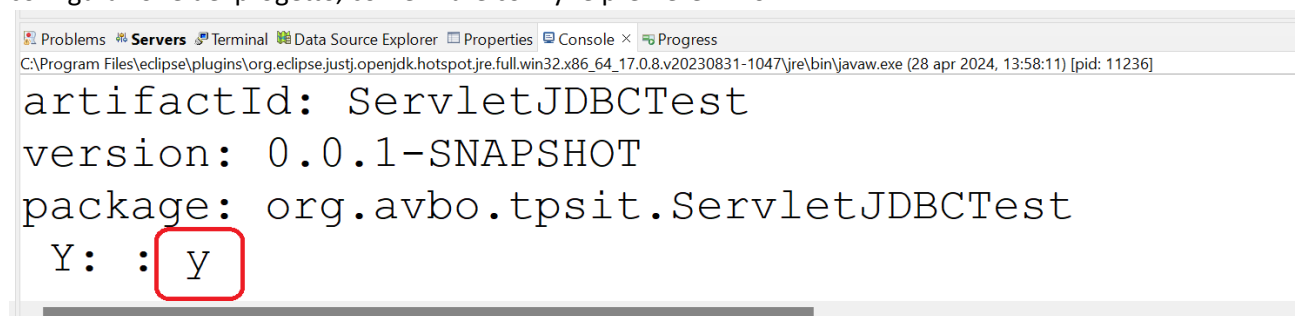
3. Al secondo passaggio del wizard di creazione del progetto Maven selezionare maven-archetype-webapp versione 1.4



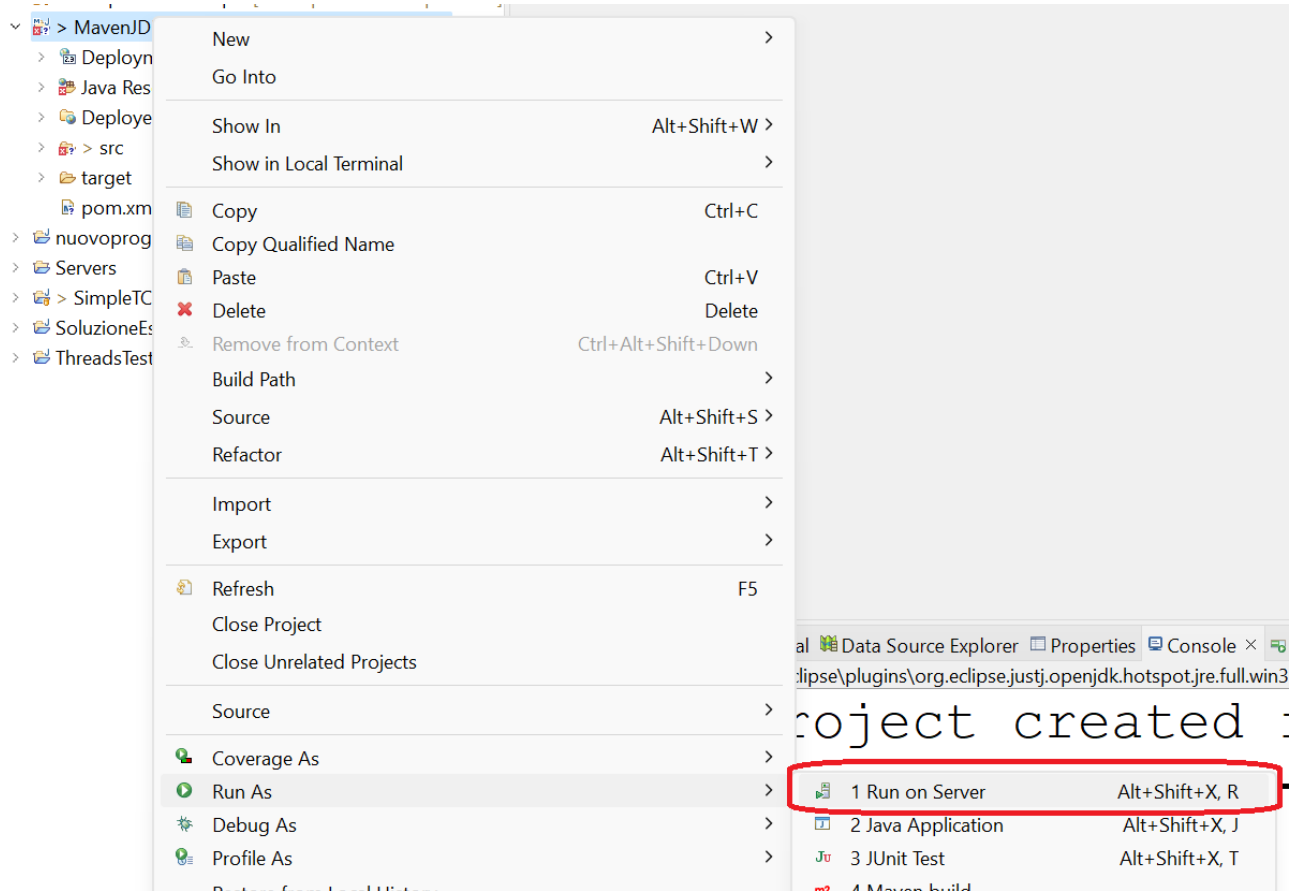
4. Terminare il wizard indicando il package e il nome del progetto.



5. Dopo il download di tutte le dipendenze nella console Maven chiederà conferma sulla configurazione del progetto, confermare con 'y' e premere invio



6. Il progetto verrà aggiunto al Workspace, provare a eseguire il progetto sul server web Tomcat (la configurazione di Tomcat non è compresa in questa guida)



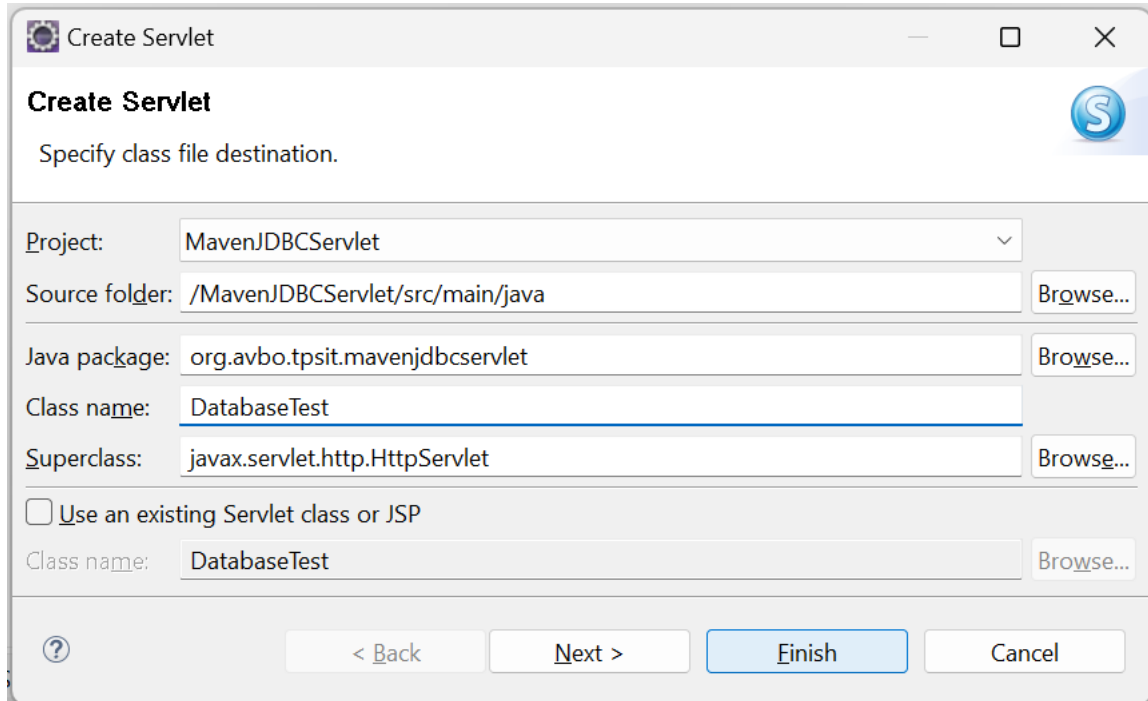
7. Dovrebbe aprirsi una pagina WEB con una scritta Hello world al centro



8. Aggiungere la dipendenza a Jakarta Servlet versione 6.0.0 al progetto



9. Creare una nuova Servlet dall'apposito menu di eclipse



The 'Create Servlet' dialog box in Eclipse IDE. It contains the following fields and options:

- Project:** MavenJDBCServlet
- Source folder:** /MavenJDBCServlet/src/main/java
- Java package:** org.avbo.tpsit.mavenjdbcervlet
- Class name:** DatabaseTest
- Superclass:** javax.servlet.http.HttpServlet
- ☐ Use an existing Servlet class or JSP
- Class name:** DatabaseTest

Buttons at the bottom: < Back, Next >, Finish, Cancel.

10. Nella nuova Servlet ci sarà un problema con gli include, correggere sostituendo i riferimenti a javax con jakarta



The code editor shows the following code for DatabaseTest.java:

```
1 package org.avbo.tpsit.mavenjdbcervlet;  
2  
3 import java.io.IOException;  
4 import javax.servlet.ServletException;  
5 import javax.servlet.http.HttpServlet;  
6 import javax.servlet.http.HttpServletRequest;  
7 import javax.servlet.http.HttpServletResponse;  
8  
9 import java.io.IOException;  
10 import jakarta.servlet.ServletException;  
11 import jakarta.servlet.http.HttpServlet;  
12 import jakarta.servlet.http.HttpServletRequest;  
13 import jakarta.servlet.http.HttpServletResponse;  
14
```

A red circle highlights the 'jakarta' package names in lines 10-13.

11. Eseguendo sul server tomcat la nuova servlet dovrebbe funzionare senza problemi



12. Fare l'override del metodo init nella Servlet facendo attenzione a non rimuovere la chiamata alla classe base

```
21
22     @Override
23     public void init() throws ServletException {
24         // TODO Auto-generated method stub
25         super.init();
26     }
27
```

13. Aggiungere la dipendenza al driver SQLite nel pom.xml

```
37     <!-- https://mvnrepository.com/artifact/org.xerial/sqlite-jdbc -->
38     <dependency>
39         <groupId>org.xerial</groupId>
40         <artifactId>sqlite-jdbc</artifactId>
41         <version>3.45.3.0</version>
42     </dependency>
43
```

14. Assicurarsi di inizializzare il driver SQLite tramite reflection fornendo la posizione della classe comprensiva di package nell'init della servlet (che viene eseguito solo una volta)

```
@Override
public void init() throws ServletException {
    super.init();

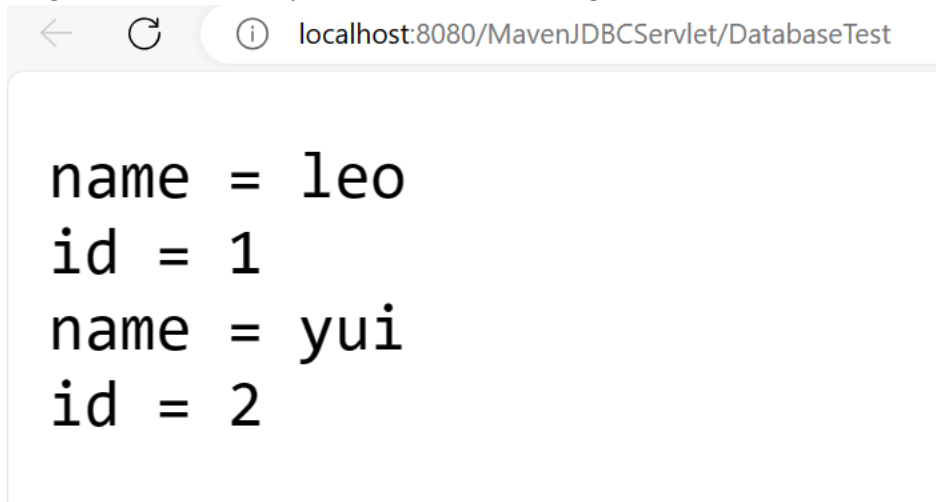
    try {
        Class.forName("org.sqlite.JDBC");
        .getConstructor().newInstance();
    } catch (InstantiationException e) {
        // TODO Auto-generated catch block
    }
}
```

15. Inserire il codice di esempio nel metodo doGet in cui viene interrogato il database utilizzando il driver scelto

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    // NOTE: Connection and Statement are AutoClosable.
    // Don't forget to close them both in order to avoid leaks.
    try {
        // create a database connection
        Connection connection = DriverManager.getConnection("jdbc:sqlite:");
        Statement statement = connection.createStatement();
        statement.setQueryTimeout(30); // set timeout to 30 sec.

        statement.executeUpdate("drop table if exists person");
        statement.executeUpdate("create table person (id integer, name string)");
        statement.executeUpdate("insert into person values(1, 'leo')");
        statement.executeUpdate("insert into person values(2, 'yui')");
        ResultSet rs = statement.executeQuery("select * from person");
        while (rs.next()) {
            // read the result set
            response.getWriter().println("name = " + rs.getString("name"));
            response.getWriter().println("id = " + rs.getInt("id"));
        }
    } catch (SQLException e) {
        // if the error message is "out of memory",
        // it probably means no database file is found
        e.printStackTrace(response.getWriter());
    }
}
```

16. Eseguire la servlet, l'output dovrebbe essere il seguente



```
name = leo
id = 1
name = yui
id = 2
```