Using JDBC

What Is JDBC in Java?

▸ JDBC (Java Database Connectivity) is a low-level API to access relational databases.

▸ Part of the Java SE platform since JDK 1.1.

▸ Provides:

  ▸ Connection, Statement, ResultSet, etc.

  ▸ SQL execution from Java code

▸ Very manual and boilerplate-heavy.

Example – Plain JDBC Code

```java
Connection conn = DriverManager.getConnection(url, user, pass);
PreparedStatement stmt = conn.prepareStatement("SELECT * FROM book WHERE id = ?");
stmt.setLong(1, 1L);
ResultSet rs = stmt.executeQuery();

if (rs.next()) {
    Book book = new Book(rs.getLong("id"), rs.getString("title"));
}
```

▸ Requires error handling, result set mapping, resource closing.

▸ Prone to verbosity and subtle bugs.

Demo 1 - Using JDBC

▸ Database with schema and data

▸ Repository with JDBC functionality

What Is Spring Data JDBC?

▸ A lightweight Spring Data project for working with relational databases.

▸ Uses JDBC under the hood but removes most boilerplate

▸ Ideal for simple domain models and performance-critical applications.

Spring Data JDBC – Benefits

▸ Spring Data JDBC gives developers full control over the SQL that is executed

▸ Integrates with the Spring ecosystem and Spring Boot with auto-configuration

Demo 2 - Using Spring Data JDBC

‣ Database with schema and data

‣ Repository with Spring Data JDBC functionality

Exercise 1 - Using JDBC

▸ Download the UsingJDBCDemo project

▸ Look at the data.sql and schema.sql and you will discover a book table

▸ Look at the com.example.exercise package for classes like BookRepository (where you will create the solution) and BookRowMapper (helper class)

▸ Uncomment one test at a time in SpringDataJDBCExerciseTests and make it compile and pass by creating methods in BookRepository

▸ Use the methods in DogRepository as guidelines