## Basic CRUD operation with DB involvement with Single entity

## Q: Develop Library management system using CURD operation with DB. Build the application only for Book with required CURD operation.

## 

## Expected output :

## 

## Follow the below steps

## 1. Prerequisites

To begin, make sure you have the following pieces of software installed on your computer:

* + JDK ([download JDK8](http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html)).
  + MySQL ([download MySQL Community Server 5.6.12](https://dev.mysql.com/downloads/mysql/)). You may also want to [download MySQL Workbench](https://dev.mysql.com/downloads/tools/workbench) - a graphical tool for working with MySQL databases.
  + JDBC Driver for MySQL ([download MySQL Connector/J 5.1.25](https://dev.mysql.com/downloads/connector/j/)). Extract the zip archive and put the mysql-connector-java-VERSION-bin.jar file into classpath (in a same folder as your Java source files).

## 

## 2. Creating a sample database

create a MySQL database called *LibraryDB* with one table *Book* .

## 3. Perform the main JDBC interfaces and classes

Use the java.sql package:

* + **DriverManager**: this class is used to register driver for a specific database type and to establish a database connection with the server via its **getConnection()** method.
  + **Connection**: this interface represents an established database connection (session) from which we can create statements to execute queries and retrieve results, get metadata about the database, close connection, etc.
  + **Statement** and **PreparedStatement**: these interfaces are used to execute static SQL query and parameterized SQL query, respectively. Statement is the super interface of the PreparedStatement interface.
  + **ResultSet**: contains table data returned by a SELECT query. Use this object to iterate over rows in the result set using **next()**method, and get value of a column in the current row using **getXXX()** methods (e.g. **getString()**, **getInt()**, **getFloat()** and so on). The column value can be retrieved either by index number (1-based) or by column name.
  + **SQLException**: this checked exception is declared to be thrown by all the above methods, so we have to catch this exception explicitly when calling the above classes’ methods.

## 4. Connecting to the database

Once the connection was established, have a Connection object which can be used to create statements in order to execute SQL queries.

## 5. JDBC Execute INSERT Statement for Books

## 6. JDBC Execute SELECT Statement for Books

## 7. JDBC Executing UPDATE Statement for Books

## 8. JDBC Execute DELETE Statement for Books