* Explain the need and benefit of ORM
  + ORM (Object-Relational Mapping), makes it easier to develop code that interacts with database, abstracts the database system, transactionality
    - ORM Pros and Cons - https://blog.bitsrc.io/what-is-an-orm-and-why-you-should-use-it-b2b6f75f5e2a
    - What is ORM? - https://en.wikipedia.org/wiki/Object-relational\_mapping

**HibernateDemo.java**

package com.example;

import javax.persistence.Entity;

import javax.persistence.Id;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

@Entity

class Book {

    @Id

    private int id;

    private String title;

    // Constructors, getters, setters

    public Book() {}

    public Book(int id, String title) {

        this.id = id;

        this.title = title;

    }

    public int getId() { return id; }

    public void setId(int id) { this.id = id; }

    public String getTitle() { return title; }

    public void setTitle(String title) { this.title = title; }

}

public class HibernateDemo {

    public static void main(String[] args) {

        // Create SessionFactory from hibernate.cfg.xml

        SessionFactory factory = new Configuration()

                .configure("hibernate.cfg.xml")

                .addAnnotatedClass(Book.class)

                .buildSessionFactory();

        // Open session and begin transaction

        Session session = factory.openSession();

        session.beginTransaction();

        // Create and save a Book

        Book book = new Book(2, "Java Programming");

        session.save(book);

        // Commit transaction and close session

        session.getTransaction().commit();

        session.close();

        factory.close();

        System.out.println("Book saved successfully!");

    }

}

**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>com.example</groupId>

  <artifactId>demo</artifactId>

  <version>1.0-SNAPSHOT</version>

  <name>demo</name>

  <!-- FIXME change it to the project's website -->

  <url>http://www.example.com</url>

  <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>junit</groupId>

      <artifactId>junit</artifactId>

      <version>4.11</version>

      <scope>test</scope>

    </dependency>

    <!-- Spring Core -->

    <dependency>

        <groupId>org.springframework</groupId>

        <artifactId>spring-context</artifactId>

        <version>5.3.29</version>

    </dependency>

    <!-- Hibernate ORM -->

    <dependency>

        <groupId>org.hibernate</groupId>

        <artifactId>hibernate-core</artifactId>

        <version>5.6.15.Final</version>

    </dependency>

    <!-- JDBC Driver (e.g., MySQL) -->

    <dependency>

        <groupId>mysql</groupId>

        <artifactId>mysql-connector-java</artifactId>

        <version>8.0.33</version>

    </dependency>

    <!-- Logging -->

    <dependency>

        <groupId>org.slf4j</groupId>

        <artifactId>slf4j-api</artifactId>

        <version>1.7.36</version>

    </dependency>

  </dependencies>

  <build>

    <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->

      <plugins>

        <!-- clean lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#clean\_Lifecycle -->

        <plugin>

          <artifactId>maven-clean-plugin</artifactId>

          <version>3.1.0</version>

        </plugin>

        <!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin\_bindings\_for\_jar\_packaging -->

        <plugin>

          <artifactId>maven-resources-plugin</artifactId>

          <version>3.0.2</version>

        </plugin>

        <plugin>

          <artifactId>maven-compiler-plugin</artifactId>

          <version>3.8.0</version>

        </plugin>

        <plugin>

          <artifactId>maven-surefire-plugin</artifactId>

          <version>2.22.1</version>

        </plugin>

        <plugin>

          <artifactId>maven-jar-plugin</artifactId>

          <version>3.0.2</version>

        </plugin>

        <plugin>

          <artifactId>maven-install-plugin</artifactId>

          <version>2.5.2</version>

        </plugin>

        <plugin>

          <artifactId>maven-deploy-plugin</artifactId>

          <version>2.8.2</version>

        </plugin>

        <!-- site lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#site\_Lifecycle -->

        <plugin>

          <artifactId>maven-site-plugin</artifactId>

          <version>3.7.1</version>

        </plugin>

        <plugin>

          <artifactId>maven-project-info-reports-plugin</artifactId>

          <version>3.0.0</version>

        </plugin>

      </plugins>

    </pluginManagement>

  </build>

</project>

**Hibernate.cfg.xml**

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

  <session-factory>

    <property name="hibernate.connection.driver\_class">com.mysql.cj.jdbc.Driver</property>

    <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/test</property>

    <property name="hibernate.connection.username">root</property>

    <property name="hibernate.connection.password">system</property>

    <property name="hibernate.dialect">org.hibernate.dialect.MySQL8Dialect</property>

    <property name="hibernate.hbm2ddl.auto">update</property>

    <property name="show\_sql">true</property>

  </session-factory>

</hibernate-configuration>

OUTPUT:

