# 8)Demonstrate integration of Hibernate with spring.

Step 1: Set up a Maven project Create a new Maven project in your preferred IDE. This will help manage the dependencies for Hibernate and Spring.

Step 2: Add Hibernate and Spring dependencies Open your project's **pom.xml** file and add the following dependencies:

<!-- Hibernate dependencies -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.6.1.Final</version>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-entitymanager</artifactId>

<version>5.6.1.Final</version>

</dependency>

<!-- Spring dependencies -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>5.3.9</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.9</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>5.3.9</version>

</dependency>

Step 3: Configure Hibernate properties Create a Hibernate configuration file, for example, **hibernate.cfg.xml**, and specify the necessary properties such as database connection details and dialect. Here's an example:

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE hibernate-configuration PUBLIC

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

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

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

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

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

<property name="hibernate.connection.username">your\_username</property>

<property name="hibernate.connection.password">your\_password</property>

<!-- Other Hibernate properties -->

</session-factory>

</hibernate-configuration>

Step 4: Create Hibernate entity classes Create Hibernate entity classes representing your database tables. Annotate them with Hibernate annotations such as **@Entity**, **@Table**, **@Id**, **@GeneratedValue**, etc.

Step 5: Configure Spring Create a Spring configuration file, for example, **spring-config.xml**, and define the necessary Spring beans for Hibernate session factory and transaction management. Here's an example:

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:tx="http://www.springframework.org/schema/tx"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context.xsd

http://www.springframework.org/schema/tx

http://www.springframework.org/schema/tx/spring-tx.xsd">

<!-- Enable component scanning -->

<context:component-scan base-package="com.example"/>

<!-- Configure Hibernate session factory -->

<bean id="sessionFactory" class="org.springframework.orm.hibernate5.LocalSessionFactoryBean">

<property name="configLocation" value="classpath:hibernate.cfg.xml"/>

<property name="packagesToScan" value="com.example.entity"/>

</bean>

<!-- Configure transaction manager -->

<bean id="transactionManager" class="org.springframework.orm.hibernate5.HibernateTransactionManager">

<property name="sessionFactory" ref="sessionFactory"/>

</bean>

<!-- Enable Spring transaction management -->

<tx:annotation-driven/>

</beans>

Step 6: Create Spring service and DAO classes Create Spring service and DAO classes to encapsulate the business logic and data access operations. Use Hibernate session factory and transactions in these classes.

Step 7: Run the application Create a Main class or a test class to bootstrap the Spring context and perform operations using the Spring services and DAOs.