**WEEK-3**

**Spring Data JPA with Spring Boot, Hibernate**

**Hands On-4: Difference between JPA, Hibernate and Spring Data JPA**

**1.pom.xml:**

<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>hibernate-example</artifactId>

    <version>1.0-SNAPSHOT</version>

    <properties>

        <maven.compiler.source>1.8</maven.compiler.source>

        <maven.compiler.target>1.8</maven.compiler.target>

    </properties>

    <dependencies>

        <dependency>

            <groupId>org.hibernate</groupId>

            <artifactId>hibernate-core</artifactId>

            <version>5.6.15.Final</version>

        </dependency>

        <dependency>

            <groupId>com.h2database</groupId>

            <artifactId>h2</artifactId>

            <version>2.1.214</version>

        </dependency>

        <dependency>

            <groupId>javax.persistence</groupId>

            <artifactId>javax.persistence-api</artifactId>

            <version>2.2</version>

        </dependency>

        <dependency>

            <groupId>org.slf4j</groupId>

            <artifactId>slf4j-api</artifactId>

            <version>1.7.36</version>

        </dependency>

        <dependency>

            <groupId>org.slf4j</groupId>

            <artifactId>slf4j-simple</artifactId>

            <version>1.7.36</version>

        </dependency>

    </dependencies>

</project>

**2.Employee.java:**

package com.example.dao;

import javax.persistence.\*;

@Entity

@Table(name = "employees")

public class Employee {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private int id;

    private String name;

    private String role;

    public Employee() {}

    public Employee(String name, String role) {

        this.name = name;

        this.role = role;

    }

}

**3.EmployeeDAO.java:**

package com.example.dao;

import org.hibernate.Session;

import org.hibernate.Transaction;

public class EmployeeDAO {

    public static void main(String[] args) {

        Employee employee = new Employee("John Doe", "Developer");

        Transaction transaction = null;

        try (Session session = HibernateUtil.getSessionFactory().openSession()) {

            transaction = session.beginTransaction();

            session.save(employee);

            transaction.commit();

            System.out.println("Employee saved successfully");

        } catch (Exception e) {

            if (transaction != null) transaction.rollback();

            e.printStackTrace();

        }

    }

}

**4.HibernateUtil.java:**

package com.example.dao;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

public class HibernateUtil {

    private static final SessionFactory sessionFactory;

    static {

        try {

            sessionFactory = new Configuration().configure().buildSessionFactory();

        } catch (Throwable ex) {

            throw new ExceptionInInitializerError(ex);

        }

    }

    public static SessionFactory getSessionFactory() {

        return sessionFactory;

    }

}

**5.hibernate.cfg.xml:**

<?xml version='1.0' encoding='utf-8'?>

<!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">org.h2.Driver</property>

        <property name="hibernate.connection.url">jdbc:h2:mem:testdb</property>

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

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

        <property name="hibernate.dialect">org.hibernate.dialect.H2Dialect</property>

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

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

        <mapping class="com.example.dao.Employee"/>

    </session-factory>

</hibernate-configuration>

**Output:**

