**Demonstrate writing HQL and Native Query**

**Entity: Country.java**

package com.example.entity;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

import lombok.\*;

@Entity

@Table(name = "country")

@Data

@Builder

@NoArgsConstructor

@AllArgsConstructor

public class Country {

@Id

private String code;

private String name;

}

**Repository: CountryRepository.java**

package com.example.repository;

import com.example.entity.Country;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Query;

import org.springframework.stereotype.Repository;

import java.util.List;

@Repository

public interface CountryRepository extends JpaRepository<Country, String> {

// HQL: Search by name using JPQL

@Query("SELECT c FROM Country c WHERE c.name = ?1")

List<Country> getCountriesByNameJPQL(String name);

// HQL: Search using LIKE

@Query("SELECT c FROM Country c WHERE c.name LIKE %?1%")

List<Country> searchCountriesByNameJPQL(String keyword);

// Native SQL: Search by name

@Query(value = "SELECT \* FROM country WHERE name = ?1", nativeQuery = true)

List<Country> getCountriesByNameNative(String name);

// Native SQL: LIKE query

@Query(value = "SELECT \* FROM country WHERE name LIKE %?1%", nativeQuery = true)

List<Country> searchCountriesByNameNative(String keyword);

}

**Test Class: CountryRepositoryTest.java**

package com.example.repository;

import com.example.entity.Country;

import org.junit.jupiter.api.Test;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.test.context.SpringBootTest;

import java.util.List;

@SpringBootTest

public class CountryRepositoryTest {

@Autowired

private CountryRepository countryRepository;

@Test

public void saveSampleCountries() {

Country c1 = Country.builder()

.code("IN")

.name("India")

.build();

Country c2 = Country.builder()

.code("US")

.name("United States")

.build();

Country c3 = Country.builder()

.code("UK")

.name("United Kingdom")

.build();

countryRepository.saveAll(List.of(c1, c2, c3));

}

@Test

public void testJPQLQuery() {

List<Country> list = countryRepository.getCountriesByNameJPQL("India");

list.forEach(System.out::println);

}

@Test

public void testNativeQuery() {

List<Country> list = countryRepository.getCountriesByNameNative("India");

list.forEach(System.out::println);

}

@Test

public void testJPQLLikeQuery() {

List<Country> list = countryRepository.searchCountriesByNameJPQL("King");

list.forEach(System.out::println);

}

@Test

public void testNativeLikeQuery() {

List<Country> list = countryRepository.searchCountriesByNameNative("King");

list.forEach(System.out::println);

}

}

**Console Output (with Hibernate logs + result)**

Hibernate: insert into country (name, code) values (?, ?)

Hibernate: insert into country (name, code) values (?, ?)

Hibernate: insert into country (name, code) values (?, ?)

Hibernate: select country0\_.code as code1\_0\_, country0\_.name as name2\_0\_ from country country0\_ where country0\_.name=?

Country(code=IN, name=India)

Hibernate: select \* from country where name = ?

Country(code=IN, name=India)

Hibernate: select country0\_.code as code1\_0\_, country0\_.name as name2\_0\_ from country country0\_ where country0\_.name like ?

Country(code=UK, name=United Kingdom)

Hibernate: select \* from country where name like ?

Country(code=UK, name=United Kingdom)