**Demonstrate Implementation of Query Methods Feature of Spring Data JPA**

**Entity: Country.java**

package com.debangshu.spring.data.jpa.entity;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

import lombok.AllArgsConstructor;

import lombok.Builder;

import lombok.Data;

import lombok.NoArgsConstructor;

@Entity

@Data

@Builder

@NoArgsConstructor

@AllArgsConstructor

@Table(name = "country")

public class Country {

@Id

private String code;

private String name;

}

**Repository: CountryRepository.java**

package com.debangshu.spring.data.jpa.repository;

import com.debangshu.spring.data.jpa.entity.Country;

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

import java.util.List;

public interface CountryRepository extends JpaRepository<Country, String> {

// Derived Query Methods

List<Country> findByName(String name);

List<Country> findByNameContaining(String nameFragment);

Country findByCode(String code);

}

**These are derived queries**, **meaning Spring will generate the SQL based on method names.**

**Test Class: CountryRepositoryTest.java**

package com.debangshu.spring.data.jpa.repository;

import com.debangshu.spring.data.jpa.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 saveCountries() {

Country india = Country.builder().code("IN").name("India").build();

Country indonesia = Country.builder().code("ID").name("Indonesia").build();

Country ireland = Country.builder().code("IE").name("Ireland").build();

countryRepository.save(india);

countryRepository.save(indonesia);

countryRepository.save(ireland);

}

@Test

public void findByExactName() {

List<Country> countries = countryRepository.findByName("India");

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

}

@Test

public void findByNameContaining() {

List<Country> countries = countryRepository.findByNameContaining("Ind");

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

}

@Test

public void findByCode() {

Country country = countryRepository.findByCode("IN");

System.out.println(country);

}

}

**Output (Console):**

Assuming saved the countries in saveCountries():

Country(code=IN, name=India)

Country(code=ID, name=Indonesia)

Country(code=IE, name=Ireland)

* findByName("India") → Returns one entry.
* findByNameContaining("Ind") → Returns India and Indonesia.
* findByCode("IE") → Returns Ireland.

**Method -> SQL**

findByName("India") -> SELECT \* FROM country WHERE name = 'India'

findByNameContaining("Ind") -> SELECT \* FROM country WHERE name LIKE '%Ind%'

findByCode("IN") -> SELECT \* FROM country WHERE code = 'IN'