**Demonstrate Implementation of O/R Mapping (Object/Relational Mapping)**

Using **Object/Relational Mapping (ORM)** features in Spring Data JPA to model a relationship between two entities using annotations like @OneToMany, @ManyToOne, etc.

**Entity: Country.java**

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

import jakarta.persistence.\*;

import lombok.\*;

import java.util.List;

@Entity

@Data

@Builder

@NoArgsConstructor

@AllArgsConstructor

@Table(name = "countries")

public class Country {

@Id

private String code;

private String name;

@OneToMany(mappedBy = "country", cascade = CascadeType.ALL, orphanRemoval = true)

private List<State> states;

}

**@OneToMany**: One country → many states.

**mappedBy** = "country": Points to the country field in the State entity.

**cascade** = ALL: Propagates save/delete operations.

**orphanRemoval** = true: Deletes child when removed from list.

**Entity: State.java**

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

import jakarta.persistence.\*;

import lombok.\*;

@Entity

@Data

@Builder

@NoArgsConstructor

@AllArgsConstructor

@Table(name = "states")

public class State {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

@ManyToOne

@JoinColumn(name = "country\_code") // foreign key

private Country country;

}

**@ManyToOne:** Many states → one country.

**@JoinColumn:** Specifies the foreign key column.

**Repository Interfaces**

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

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

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

public interface CountryRepository extends JpaRepository<Country, String> {

}

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

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

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

public interface StateRepository extends JpaRepository<State, Long> {

}

**Test Class: CountryRepositoryTest.java**

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

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

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

import org.junit.jupiter.api.Test;

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

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

import java.util.Arrays;

@SpringBootTest

public class CountryRepositoryTest {

@Autowired

private CountryRepository countryRepository;

@Test

public void saveCountryWithStates() {

State state1 = State.builder().name("West Bengal").build();

State state2 = State.builder().name("Maharashtra").build();

Country country = Country.builder()

.code("IN")

.name("India")

.states(Arrays.asList(state1, state2))

.build();

// Set reverse relationship

state1.setCountry(country);

state2.setCountry(country);

countryRepository.save(country);

}

}

**Output:**

Hibernate: insert into countries ...

Hibernate: insert into states ...

**Database Tables:**

countries: code = IN, name = India

states: (id = 1, name = West Bengal, country\_code = IN), etc.