**Demonstrate implementation of O/R Mapping**

***Step 1: Country Entity***

import jakarta.persistence.\*;

import java.util.List;

@Entity

public class Country {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

@OneToMany(mappedBy = "country", fetch = FetchType.LAZY, cascade = CascadeType.ALL)

private List<State> states;

@ManyToMany(fetch = FetchType.EAGER)

@JoinTable(

name = "country\_language",

joinColumns = @JoinColumn(name = "country\_id"),

inverseJoinColumns = @JoinColumn(name = "language\_id")

)

private List<Language> languages;

}

## *Step 2: State Entity*

import jakarta.persistence.\*;

@Entity

public class State {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

// Many States belong to one Country

@ManyToOne(fetch = FetchType.LAZY)

@JoinColumn(name = "country\_id")

private Country country;

}

## *Step 3: Language Entity*

import jakarta.persistence.\*;

import java.util.List;

@Entity

public class Language {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

@ManyToMany(mappedBy = "languages", fetch = FetchType.LAZY)

private List<Country> countries;

}

Repositories

public interface CountryRepository extends JpaRepository<Country, Long> {}

public interface StateRepository extends JpaRepository<State, Long> {}

public interface LanguageRepository extends JpaRepository<Language, Long> {}

## *Example JSON for Creating a Country*

{

"name": "India",

"states": [

{ "name": "Tamil Nadu" },

{ "name": "Kerala" }

],

"languages": [

{ "name": "Tamil" },

{ "name": "Hindi" }

]

}