**Date : 17/10/2020**

**Spring Boot 7AM**

**Mr. RAGHU**

**-----------------------------------------**

**Spring Boot - Data JPA Association Mapping**

=> 2 Model classes

=> HAS-A Relation between them

=> Check for Collection Type or not?

=> Provide multiplicity annotation

=> Provide Join Column | Join Table

=> Define Repositories

=> Define Runner class for data insert

> create objects

> link based on relation

> save into database

=> Check for tables for output.

=====================code=====================

Many-To-One : \*...1

Employee----<> Dept

HAS-A

1. Create Spring Starter Project

Name : SpringBoot2DataJpaManyToOneEx

Dep : Spring Data Jpa, lombok, mysql

2. application.properties

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://localhost:3306/boot7am

spring.datasource.username=root

spring.datasource.password=root

spring.jpa.show-sql=true

spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect

spring.jpa.hibernate.ddl-auto=create

3. Model class

package in.nareshit.raghu.model;

import javax.persistence.Entity;

import javax.persistence.Id;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.NoArgsConstructor;

@Data

@NoArgsConstructor

@AllArgsConstructor

@Entity

public class Dept {

@Id

private Integer id;

private String code;

}

------------------

package in.nareshit.raghu.model;

import javax.persistence.Entity;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.ManyToOne;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.NoArgsConstructor;

@Data

@NoArgsConstructorAssociationMapping\_Overview

@AllArgsConstructor

@Entity

public class Employee {

@Id

private Integer id;

private String name;

private Double sal;

// \*...1

//Employee---<> Dept

@ManyToOne

@JoinColumn(name="didFk")

private Dept dob;

}

-------------------------

4. Repository Interface

package in.nareshit.raghu.repo;

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

import in.nareshit.raghu.model.Dept;

public interface DeptRepository

extends JpaRepository<Dept, Integer> {

}

-------------

package in.nareshit.raghu.repo;

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

import in.nareshit.raghu.model.Employee;

public interface EmployeeRepository

extends JpaRepository<Employee, Integer> {

}

5. Runner class

package in.nareshit.raghu.runner;

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

import org.springframework.boot.CommandLineRunner;

import org.springframework.stereotype.Component;

import in.nareshit.raghu.model.Dept;

import in.nareshit.raghu.model.Employee;

import in.nareshit.raghu.repo.DeptRepository;

import in.nareshit.raghu.repo.EmployeeRepository;

@Component

public class DataInsertRunner implements CommandLineRunner {

@Autowired

private DeptRepository drepo;

@Autowired

private EmployeeRepository erepo;

@Override

public void run(String... args) throws Exception {

Dept d1 = new Dept(550, "DEV");

Dept d2 = new Dept(551, "BA");

Employee e1 = new Employee(10, "A", 3000.0, d1);

Employee e2 = new Employee(11, "B", 4000.0, d1);

Employee e3 = new Employee(12, "C", 5000.0, d2);

Employee e4 = new Employee(13, "D", 6000.0, d2);

//save objects

drepo.save(d1);

drepo.save(d2);

erepo.save(e1);

erepo.save(e2);

erepo.save(e3);

erepo.save(e4);

}

}

=====================code=====================

One-To-Many : 1...\*

Vendor----<> Contract

HAS-A

1. Model

package in.nareshit.raghu.model;

import javax.persistence.Entity;

import javax.persistence.Id;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.NoArgsConstructor;

@Data

@NoArgsConstructor

@AllArgsConstructor

@Entity

public class Contract {

@Id

private Integer cid;

private String code;

private Double bidAmt;

private String ctype;

}

package in.nareshit.raghu.model;

import java.util.List;

import javax.persistence.Entity;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.OneToMany;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.NoArgsConstructor;

@Data

@NoArgsConstructor

@AllArgsConstructor

@Entity

public class Vendor {

@Id

private Integer vid;

private String vname;

private String vaddr;

/\*\*

\* 1...\*

\* Vendor ---<> Contract

\* HAS-A

\*/

@OneToMany

@JoinColumn(name="vid\_fk")

private List<Contract> cobs; //HAS-A

}

2. Repository Interfaces

package in.nareshit.raghu.repo;

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

import in.nareshit.raghu.model.Contract;

public interface ContractRepository

extends JpaRepository<Contract, Integer> {

}

package in.nareshit.raghu.repo;

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

import in.nareshit.raghu.model.Vendor;

public interface VendorRepository

extends JpaRepository<Vendor, Integer> {

}

3. Runner class

package in.nareshit.raghu.runner;

import java.util.Arrays;

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

import org.springframework.boot.CommandLineRunner;

import org.springframework.stereotype.Component;

import in.nareshit.raghu.model.Contract;

import in.nareshit.raghu.model.Vendor;

import in.nareshit.raghu.repo.ContractRepository;

import in.nareshit.raghu.repo.VendorRepository;

@Component

public class DataInsertRunner implements CommandLineRunner {

@Autowired

private VendorRepository vrepo;

@Autowired

private ContractRepository crepo;

@Override

public void run(String... args) throws Exception {

Contract c1 = new Contract(550, "C1", 5000.0, "ROAD EXT");

Contract c2 = new Contract(551, "C2", 6000.0, "WATER PIPES");

Contract c3 = new Contract(552, "C3", 7000.0, "CLEAN STREETS");

Vendor v1 = new Vendor(8104, "SAM", "HYD", Arrays.asList(c1,c2,c3));

crepo.save(c1);

crepo.save(c2);

crepo.save(c3);

vrepo.save(v1);

}

}

----------------------------------------------------------------------------

Q) What is CascadeType in Hibernate? Why it is used? Explain?

A)

Q) What is the difference b/w Association, Composition and Aggregation?

A)

Q) In place of List, can we use Set type at Association Mapping?

Which one good? Why?

A)

----------------------------------------------------------------------------