Date : 04/02/2021
Spring Boot 9AM
Mr. RAGHU

Data JPA : Association Mapping

Many-To-Many:-
One extra table is created ie called as JoinTable
with 2 columns - JoinColumn, InverseJoinColumn.

```
               *...*
       Student ---<> Course
              HAS-A
```

----Association Mapping Implementation Steps---

a) write Two Model classes and apply
HAS-A Relation between Model classes

b) Check For Collection/Non-Collection Type
if Collection type then modify HAS-A Variable
as Collection variable.

c) Apply Multiplicity Annotation

1...1	@ManyToOne + Unique
1...*	@OneToMany
*...1	@ManyToOne
...	@ManyToMany

d) Provide JoinColumn (FK Column) or JoinTable(JoinColumn + JoinColumn)

---

code

1. Models

```
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 Course {
	@Id
	private Integer cid;
	private String cname;
	private Double cfee;

}
-------
package in.nareshit.raghu.model;
```

```
import java.util.List;

import javax.persistence.CascadeType;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.JoinTable;
import javax.persistence.ManyToMany;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
@Entity
public class Student {
        @Id
        private Integer sid;
        private String sname;
        private String sloc;

        @ManyToMany(cascade = CascadeType.ALL, fetch = FetchType.EAGER)
        @JoinTable(name="stdcrstb",
                joinColumns = @JoinColumn(name="sidFk"),
                inverseJoinColumns = @JoinColumn(name="cidFk")
        )
        private List<Course> cobs; //HAS-A
}
```

2) Repo

```
package in.nareshit.raghu.repo;

import org.springframework.data.jpa.repository.JpaRepository;
import in.nareshit.raghu.model.Student;

public interface StudentRepo
        extends JpaRepository<Student, Integer> {

}
------------
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 Course {
        @Id
        private Integer cid;
        private String cname;
        private Double cfee;

}
```

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.Course;
import in.nareshit.raghu.model.Student;
import in.nareshit.raghu.repo.CourseRepo;
import in.nareshit.raghu.repo.StudentRepo;

@Component
public class DataInsertRunner implements CommandLineRunner {
        @Autowired
        private CourseRepo crepo;

        @Autowired
        private StudentRepo srepo;

        @Override
        public void run(String... args) throws Exception {
                Course c1 = new Course(58, "JAVA", 5000.0);
                Course c2 = new Course(59, "UI", 4000.0);
                Course c3 = new Course(60, "AWS", 5500.0);

                /*crepo.save(c1);
                crepo.save(c2);
                crepo.save(c3);*/


                Student s1 = new Student(10, "A", "HYD",
Arrays.asList(c1,c2));
                Student s2 = new Student(11, "B", "CHN",
Arrays.asList(c2,c3));

                srepo.save(s1);
                srepo.save(s2);

        }
}
----------------------------------------------------------------------
```